80 U.S.

Vol IV Number 2 - $3.00 per Copy

THE TRS·80 USERS JOURNAL

Mar/Apr 1981

B1 (V) .6*B2 C

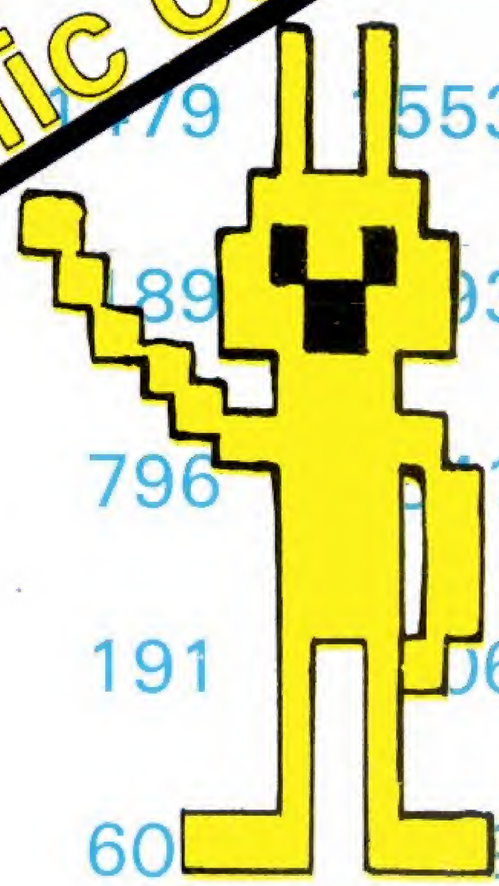
21

A B C D E F

1	MONTH	JAN	FEB	MAR	APR	MAY
2	SALES	2130	2236	2348	2465	2529
3	COST OF	1278	1341	1409	1479	1553
4	ADMINISTR	179	182	186	189	193
5	INCOME BE	678	712	753	796	841
6	INCOME TA	161	170	180	191	206
7	NET INCOM	511	541	572	60	9

VISICALC

The visible electronic calculator

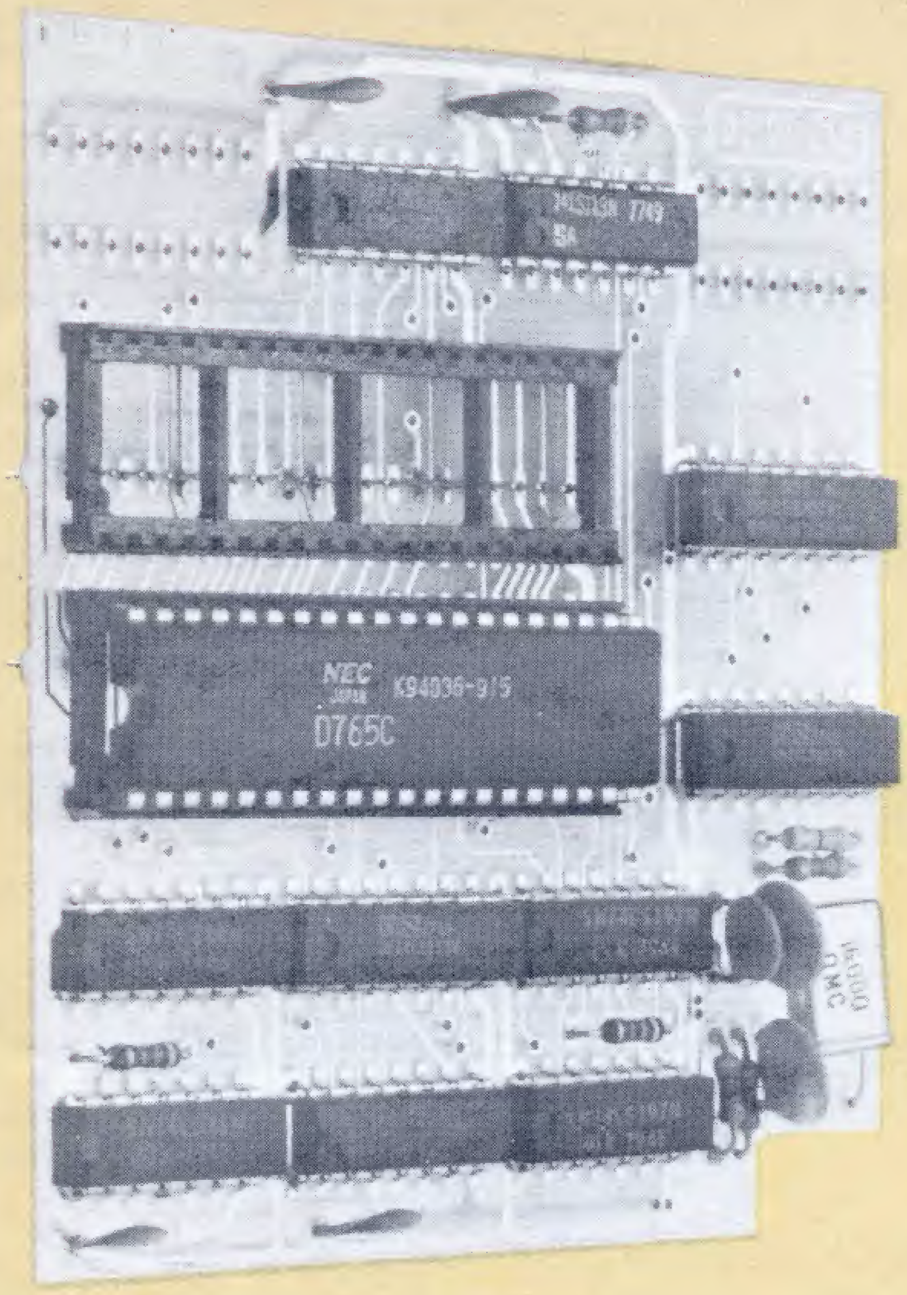


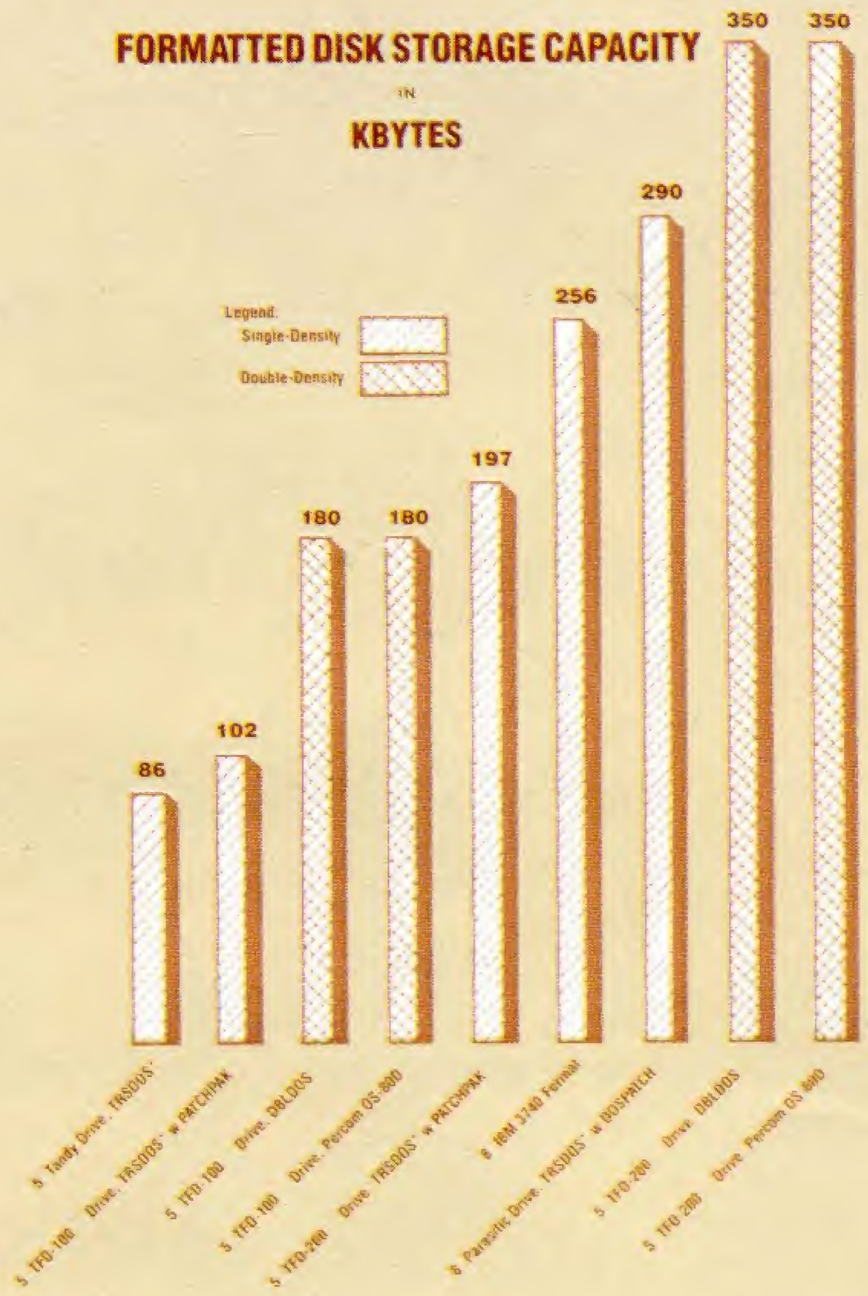
In This Issue:

Bill Barden, Jr. Talks about The New Color Computer

TRS-80* Model I Computer Owners . . .

Store Up to 350 Kbytes on a 5″ Disk

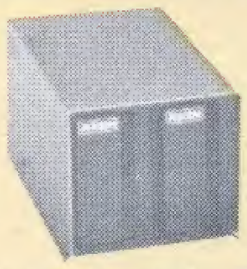




The DOUBLER™. It packs almost twice the data on a disk track as your single-density system. Depending on the type of drive, you can store up to **four** times more data on one side of a minidiskette than you can store using a standard Model I mini-disk drive.

- The DOUBLER™ reads, writes and formats **either** single- or double-density minidiskettes.
- Proprietary design allows you to continue to run TRSDOS*, NEWDOS‡, Percom OS-80™ or other single-density software **without making any changes** to software or hardware. Switch to double-density operation at any convenient time.
- Includes DBLDOS™, a **TRSDOS* compatible** double-density disk operating system.
- CONVERT utility, on DBLDOS™ minidiskette, converts files and programs from single- to double-density or double- to single-density.
- The DOUBLER™ circuit card **includes high performance data separator, write precompensation** circuits for reliable disk read operations — even with 80-track drives.
- **Plug-in Installation** — The DOUBLER simply plugs into the disk controller socket of your Expansion Interface, requiring no strapping or trace cutting. Expansion Interface disk controller may be completely restored to original configuration by simply removing the DOUBLER™ and re-installing the original disk controller chip.
- Works with standard 35-, 40-, 77- and 80-track mini-disk drives rated for double-density operation.
- Introductory price, including DBLDOS™ and format conversion utility on minidiskette, **only** $219.95.

Mini-Disk Systems



More storage capacity, higher reliability — from Percom, the industry leader. One-, two- and three-drive configurations in either 40- or 77-track format. Fully burned-in. From only $399.

Double-Density Software

(On diskette — with instruction manual.)

OS-80D™ Double-Density Disk Operating System — This double-density upgrade version of Percom's acclaimed OS-80™ resides *entirely* in RAM — requiring only 7.5-Kbytes! A BASIC programmer's "dream operating system," even utilities are in BASIC.

DOUBLEZAP-II/80 This program modifies Apparat's NEWDOS/80‡ to run either double — or single-density programs — even to run a mix of the two formats on one system!

DOUBLEZAP-II/V This program modifies Virtual Technology's VTOS 4.0†† to provide the same capability as DOUBLEZAP-II/80 provides for NEWDOS/80.

Call toll-free, **1-800-527-1592**, for the address of your nearest authorized Percom dealer, or to order directly from Percom.

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

Settle for More from Your TRS-80

BASIC Compiler. With TRS-80 BASIC Compiler, your Level II BASIC programs will run at record speeds! Compiled programs execute an average of 3-10 times faster than programs run under Level II. Make extensive use of integer operations, and get speeds 20-30 times faster than the interpreter.

Best of all, BASIC Compiler does it with BASIC, the language you already know. By compiling the same source code that your current BASIC interprets, BASIC Compiler adds speed with a minimum of effort.

And you get more BASIC features to program with, since features of Microsoft's Version 5.0 BASIC Interpreter are included in the package. Features like the WHILE . . .WEND statement, long variable names, variable length records, and the CALL statement make programming easier. An exclusive BASIC Compiler feature lets you call FORTRAN and machine language subroutines much more easily than in Level II.

Simply type in and debug your program as usual, using the BASIC interpreter. Then enter a command line telling the computer what to compile and what options to use.

Voila! Highly optimized, Z-80 machine code that your computer executes in a flash! Run it now or save it for later. Your compiled program can be saved on disk for direct execution every time.

Want to market your programs? Compiled versions are ideal for distribution.* You distribute only the object code, not the source, so your genius stays fully protected.

BASIC Compiler runs on your TRS-80 Model I with 48K and disk drive. The package includes BASIC Compiler, linking loader and BASIC library with complete documentation. $195.00.

*Microsoft royalty information for the sale of programs compiled with BASIC Compiler is available from Microsoft.

muMATH Symbolic Math System expands your TRS-80 beyond the limits of numerical evaluation to a much higher level of math sophistication.

Symbolic mathematics is muMATH's power. For the first time, algebra, trigonometry, calculus, integration, differentiation and more can be performed on a system smaller than an IBM 370. And in a fraction of the time you could do them manually.

Yet for all its power, muMATH is simple to use.

To perform a differentiation you could enter: ?DIF (A*X ↑ 3 + SIN(X ↑ 2),X);

In almost no time, the computer would reply with: @2*X*COS(X ↑ 2) + 3*A*X ↑ 2.

Or to add fractions: ?1/3 + 5/6 + 2/5 + 3/7;

The instantaneous answer: 419/210.

Or to perform a more difficult trigonometric expansion you enter: SIN(2*Y)*(4*COS(X)↑3—COS(3*X) + SIN (Y)*(COS(X+Y+#PI) — COS(X—Y));

Just a few seconds later, the computer replies: @4*SIN(Y)*COS(X)*COS(Y).

muMATH has virtually infinite precision with full accuracy up to 611 digits.

If you use math, you'll find countless ways to save time and effort with muMATH. It's a professional tool for engineers and scientists. A learning tool for students at any level from algebra to calculus.

And if you want to expand your capabilities even beyond the standard muMATH, the option is open. muSIMP, the programming language in which muMATH is written, is included in the muMATH package. A superset of the language LISP, muSIMP is designed especially for interactive symbolic mathematics and other artificial intelligence applications.

muMATH and muSIMP were written by The Soft Warehouse, Honolulu, Hawaii. Priced at $74.95, the package includes muMATH, muSIMP and a complete manual. It requires a Model I TRS-80 with 32K and single disk. muMATH for the Apple II Computer will be available later this year.

You can buy muMATH and BASIC Compiler at computer stores across the country that carry Microsoft products. If your local store doesn't have them, call us. 206-454-1315. Or write Microsoft Consumer Products, 400 108th Ave. N.E., Suite 200, Bellevue, WA 98004.

MICROSOFT
CONSUMER PRODUCTS

The emergence of the small computer has been viewed by many small businesses as either a blessing or an unnecessary expense. Many small businessmen will consider using a computer only because they think that the competition will be using one, and they feel they need to keep up. Hence, the feeling that it is an expense.

Others may think that using a small computer in their business may eliminate the need for additional personnel, and therefore reflects an overall saving in their operation.

Both viewpoints, of course, contain very little truth. While it is true that the competitor down the street may have better management information because of the use of his computer, it certainly will not eliminate the need for more employees. In fact, if he uses his computer properly, he may just find that an increase of business may require more employees.

A large segment of current business computer users have one simply because they are interested in the novelty of using it. These users represent a large portion of those who currently use one successfully in their business. They are successful only because they are interested in it, and have taken the time to dig in and find out more about computers and how they work. These are the same people who will buy utility programs, try different programming languages, and make changes to off-the-shelf software for their own use.

But how about those hundreds of thousands of small businesses who could, but do not use a computer? These are people who are not interested in the novelty, but simply want to buy a system, plug it in, and let their "girl" operate it, just like they did when they got their Xerox machine.

Let's take an example from real life: Today I went to my Doctor for an annual physical. The Doctor shares a new building with two or three other Doctors, and they own the building jointly and share limited lab facilities. His receptionist was opening mail and hand posting accounts. Later I saw her running statements on the copy machine. During the course of the exam, I had to go to the lab for a blood sample. Again, the lab was using card files. Following this was a trip across the street to a Radiological office for a chest X-ray. There, I was asked if I had had an X-ray in the last five years. I had, so the clerk went to a wall full of file drawers and found a 3 X 5 card with my information on it. Later, I went to a large drug store to have a prescription filled. The drug store, in spite of the fact they have three or four Pharmacists filling prescriptions all day long, still sends out hand posted statements.

Why are these people still doing hand posting and keeping card files? Do they even know about computers? My own Doctor's reaction to these questions was typical of many others I have asked. Yes, they know that small computers are now available for routine office tasks. But they don't think that computer hardware and software has it all together yet, and they are waiting...waiting for that all-inclusive system that does it all with the push of a button.

Perhaps the power of the small computer has been oversold. The ads on TV indicate that it is the end-all of accounting problems. Yet, when the small businessmen get down to the details, they find it really is not, and that they need to become much more involved in it than they did with their office copier. Further, the system they now use works, and they are fearful of having to stop the Merry-Go-Round to change the horses.

Small business is not like a large corporation, where a team can be assigned to oversee the changeover and still another team to implement the procedures. To a small business, installing even an Apple or TRS-80 can easily be a time of chaos and confusion.

It's not their fault. The computer must be sold for what it really is. The software must be complete and tailored to the business. During that crucial time of changeover, someone must be there to give assistance and advice. You need to know the businessman's needs if you are to sell him, and you have to know your own hardware and software first.

Selling hardware and software out of a store or by mail-order works only if the customer is already pre-sold and is interested because of the novelty of owning a computer. The rest are out there... waiting for you to come to them.

Mike

80-U.S.

JOURNAL

Editor/Publisher
I Mike Schmidt
Sales/Promotion
Margaret Farrell
Managing Editor
Tom Huber
Associate Editor
Terry R Dettman

Contributing Editors
Phil Pilgrim
Larry S Panattoni
Bill Schroeder
Spencer Hall
R C Bahn

Art/Layout/Design
Grace McNamara
Circulation
Margaret Murray
Typesetting
Jill Blackburn
Reviewers
Cameron C Brown
Pat Perez
W W Harper II

Printed by
Peninsula Gateway Press
Gig Harbor, Washington

DISTRIBUTORS
Europe
Hofacker Verlag
Tegernseer Strasse 18
D-8150 Holzkirchen/Obb
West Germany
United Kingdom
The Software House
146 Oxford Street
London, W.1 England

The Cover

Our cover this issue is the work of our resident art director, and shows a representation of the use of VisiCalc.

ISSN Publication #0199-1035

Published bimonthly in the months of January, March, May, July, September and November by **80-Northwest Publishing Inc 3838 South Warner Street, Tacoma, Washington 98409**

Subscription price in the United States is $16.00 for one year, $31.00 for two years and $45.00 for three years.

Foreign Subscriptions (except Canada): $30.00 for one year only, remitted in U.S. funds please.

Canada: $20.00 for one year only.

All issues, except United States, Canada and Central America are sent via Airmail.

Printed in the United States of America

POSTMASTER: Please send change of address form 3579 to 80-U.S. Journal, 3838 South Warner Street, Tacoma, WA 98409

Second Class Postage PAID at Tacoma, Washington

80-U.S.

THE TRS-80 USERS JOURNAL

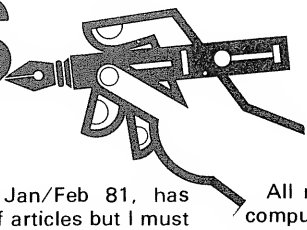
*TRS-80 is a trademark of the Tandy Corporation

Volume IV Number 2 March/April 1981

IN THIS ISSUE

Departments

LETTERS



Have just bought a copy of 80-U.S. I am glad to discover another source of information on how to make better use of my newly purchased TRS-80.

Unfortunately, however, your journal didn't help me as much as it could... You see, I own a TRS-80 Color Computer.

Now, the problem with the Color Computer is that it runs on a Motorola MC6809E microprocessor.

Enough said?

So, what are we going to do? I want to expand my newly gained knowledge of programming techniques beyond color Basic, but everywhere I turn, what do I see? Booklets on the Z-80, articles on the Z-80, manuals on the Z-80.

Nothing on the 6809E.

Anyway, I applied some good ol' "Murican" initiative to the problem and located the source of info on the 6809E: Motorola Semiconductors in Austin, Texas! A quick phone call netted me their data sheet (entitled "Advance Information," by the way! -and printed just last August!) on the new chip. In case you're interested, the number to call is (512) 928-6800 and ask for applications.

So now I have a data sheet. The essential info! But--I'm a novice programmer. Yep, must admit it: I'm wet behind the ears! However, I'm eager to learn, and very frustrated that no one (not even Radio Shack's computer personnel) seems to know much about how the 6809E is structured. So here I sit, reading all of this wonderful "TRS-80" material, article after article, which, however, I can't apply to my TRS-80 machine! I'm really beginning to feel that I own the "black sheep" of the TRS-80 family. Nothing will work on it. NOTHING (well, hardly anything) that has been designed for the other TRS-80 products. Any program with PEEK/POKE instructions, or graphics, or assembly language is not compatible. The graphics characters are very different; the pixel configuration is different (2x2 instead of 3x2); the memory storage is "reversed." And I don't know that much about programming in assembly language to make the needed changes easily.

So, I come to the point of this letter. Would you persuade Jim Crocker to begin (SOON) a series of articles on programming the 6809E in assembly language? And would you solicit articles geared for the Color Computer hobbyist? Thanks!

Gordon F. Ross
San Francisco, CA

(As was mentioned in our last issue, Jim is no longer with our staff, however, see the feature story in this issue covering the color computer. We expect to see more like it in future issues. Ed)

Your latest issue, Jan/Feb 81, has another great bunch of articles but I must take issue on the one about NEWDOS80 and VTOS 4.0. I was one of the first to get the NEWDOS80 and was then using NEWDOS-PLUS. Pete Carr stated that the ZAP sheets meant no waiting or calling on the telephone but that is not truel I have sent in the registration card, called Apparat, written them and still cannot get the new ZAP sheets. It probably would be better to pirate a recent copy which I presume would have corrections already made but I am trying to use this the legal way. It is a great DOS but the service from these people is horrible and this should also be included in your reviews. I even paid $149.50 not knowing about the rebate for previous NEWDOS owners.

I use the NEWDOS80 more than VTOS 4.0 for many of the same reasons Pete mentioned but am starting to use the VTOS only because I feel that there may be some problems with NEWDOS80 and without anyone to turn to, I may lose some file, etc.

The major problem I have found with the super copy commands is the CBF (copy by file) when used with a disk having the system programs on it. Some functions do not operate properly on the new disk and I suspect some programs look at a specific location on the disk for the system program and the CBF feature copies in the next available space except for BOOT/SYS and DIR/SYS.

One other problem was that some of my programs poked the clock on/off values into 17676-17677 rather than using the CMD function. (I never like to run the disk drive if not necessary.) It appears that some changes were made in this area and now the correct pokes are 17910,242 for clock on, and 17910,201 for clock off.

I enjoy your magazine and look forward to each issue. Keep up the good work.

Norman Dale Roder
Chesterfield, MO

cc: Apparat, Inc

(Apparat replies):

Dear Mr Roder:

The copy of your letter of December 26, 1980 to 80-U.S. Journal has been received.

You have not been sent any ZAP sheets for your copy of NEWDOS80 as none have been mailed to any user since receipt of your registration card in the third week of December 1980. You stated on the registration card that you had purchased NEWDOS80 "about May". The registration card is postmarked December 15th, 1980 and was received after that date.

All mailings to users are made from a computer mailing list prepared using names and addresses obtained from the returned registration cards. There is no other way in which we could have had a name or address to which to direct ZAP sheets to you. You purchased your copy of NEWDOS80 from a dealer, therefore, we do not have even an invoice bearing your name and address.

In delaying the return of your registration card for some seven months, you have created your own delay in receiving the continuing support offered by Apparat. Surely you cannot reasonably expect us to accept that responsibility.

If you are having difficulties of any nature with any of the NEWDOS80 utilities, or if you have encountererd a flaw in the system, the method for receiving assistance is detailed in the documentation.

Your name and address have been added to the mailing list and you will receive ZAP sheets and other mailings as they are made available to all users. If I may assist you with any problem, please feel free to contact me personally.

A C McFarling
Apparat, Inc

I have just gone through a very intensive course in TRS-80 programming. I borrowed 11 back issues of 80-U.S. for a week. I spent my free time during the day copying hints and tidbits and my nights, all night some nights, entering programs. I can't begin to tell you how much I've learned about the TRS-80 and programming and assembly language.

I think my favorite is Phil Pilgrim and his System/Command column. (Love the "Software Lower Case", Jul/Aug 79) Thanks to him I took the plunge into assembly language.

One question though. What is the reasoning behind leaving out every darn space in a program that is meant to be copied? (A Calendar Maker, Jan/Feb 81) I added the spaces and it still fit in my 16K Level II! Other than that it ran great and helps me demonstrate my computer.

Also, my compliments to Louise Frankenberg (Hexdump, Mar/Apr 80). I think she meant it to be a utility but with her own modifications in the next issue it bacame a very educational routine.

In case I've been too subtle, what I'm trying to say is you people are fantastic!! Keep up the outstanding work!

Dave Bower
Virginia Beach, VA

(Louise Frankenberg's modifications appeared in the Sep/Oct 80 issue, not in the one following Mar/Apr 80 as you suggest. Concerning spaces in programs,

(Continued on Page 6)

This printer costs less than $450. Beat that... if you can.

Epson.

This is the Epson MX-70. The lowest priced dot matrix printer you can buy. Now, that in itself should make it very attractive to a lot of people. But you ain't heard the half of it.

To begin with, the MX-70 has a lot more in common with our now-famous MX-80 than just the name. Like unequalled Epson reliability. And technological breakthroughs like the world's first disposable print head. But frankly, the MX-80 packs a lot more power than some people need. So we built the MX-70 to be a no-frills printer. At a no-frills price.

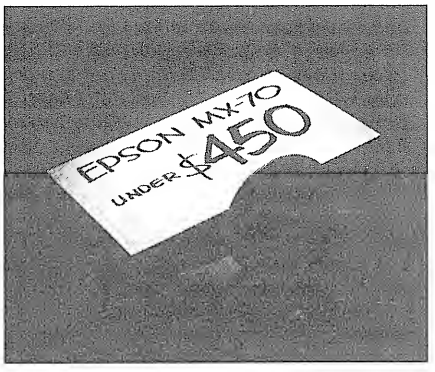
But the MX-70 is still a great little printer. We give you 80 CPS unidirectional printing. Top-of-form recognition. Programmable line feed and form lengths. Plain paper printing. An easy-to-read 5x7 matrix. Self test. And an adjustable tractor feed.

That's what you'd expect from a basic little printer. But here's something you wouldn't expect: the finest graphics package on the market today. Free.

We call it GRAFTRAX II. And it means 480 dots across the page, resolution to 60 dots per inch, and a graphic image free of the jitter and overlap that plagues other printers. You get cleaner grays and finer point resolution.

So now you've got a choice. You want more power and extra functions, you buy the MX-80. You want a basic little printer that prints, and keeps on printing, you buy the MX-70. They're both at your dealer now.

But at this price, you'd better hurry.



EPSON
EPSON AMERICA, INC.

23844 Hawthorne Boulevard • Torrance, California 90505 • (213) 378-2220

we have found that it is best to leave an author's program intact. Even renumbering in 10's sometimes causes problems (when the comma is used instead of THEN for example). Most of our programs are not typeset, but run off on a line printer from the original, working version and photographed. Ed.)

I was really thrilled to see my article, "A Basic Memory Saver" in your Nov/Dec 80 issue. I especially appreciate the fact that very little editing was done to it. I do have a comment about one sentence that was added to the text. The one that warns the reader about an ?OM error message after running the program. The purpose of including the CLEAR command in the instructions is to eliminate that very problem. Try it. I believe that the CLEAR command also resets the other pointers so that the two instructions that were added to the program for that purpose are probably not necessary.

I also thought I would pass along a couple of ROM routines I discovered:

2B7E - Translates a Basic line of text from internal to LIST format. (Command codes are converted to the actual command names). On input, HL points to the command text to be translated. (The text must end with a zero byte). Translated text is stored at the location indicated by the word at 40A7H.

2B75 - Prints a line (pointed to by HL) to the screen. End of line indicated by a zero byte. 28A7 has been mentioned for this function, but that routine will not display double quotes (22H), this one will.

Keep up the great work on your mag. It gets better every time.

Peter A Lewis
Dallas, TX

(Some machines apparently give the ?OM error, ours must be one of them as we have experienced it on three different Model I's with several programs. Ed)

It should be of interest that IF-THEN-ELSE statements may be nested in a similar manner to FOR-NEXT statements. This is shown in the following sample program:

```
100 FOR J=0 TO 10
200 A=RND(5) : B=A*A
300 IF A <> 1 THEN
      IF A <> 2 THEN
      IF A <> 3 THEN
      400
      ELSE B=30
      ELSE B=20
      ELSE B=10
310 PRINT"AT LINE 310 A=";A,"B=";B
320 GOTO 410
400 PRINT"AT LINE 400 A=";A,"B=";B
410 NEXT J
```

Robert A Hood
Bremerton, WA

I always enjoy my subscription to your fine journal, however there was one article that seems to have a discrepancy with my copy of NEWDOS.

In your Jan/Feb 81 edition, on page 120, there is an article entitled "Break-With JKL". In the first suggested method there was a zap provided in SYS1/SYS on track 10, sector 4, beginning with byte 8A, modifying a series of bytes starting with 02 02 00 4E, etc.

I have a late issue of NEWDOS+ (4/16/80) and am enclosing a listing of track 10. There is a totally different series beginning at 8A, but the series that was pointed out seems to begin at B9. Making the modification at B9 resulted in line feeds with the JKL activation but nothing else.

Making the modification at 8A was even more useless. The DOS would boot and write a command, but as soon as ENTER is pressed there is an immediate return to DOS READY.

Similarly, the 2nd, assembled program when run produced a full-screen hatch pattern. It seems fairly obvious that Apparat has issued restructured NEWDOS+ disks at some point in time. I am writing this mainly to inform you of the above as I am midway into the documentation on NEWDOS80 and will switch over to that operating system as soon as I can.

Roger Bass
Carlsbad, CA

In the Nov/Dec 80 issue of 80-U.S. there is a letter from a D D Freeman telling how to convert a decimal to a hexadecimal number on the TRS-80. I must assume Mr Freeman is using a TRS-80 Model I, since my Model II has the statement HEX$(number) for decimal to hex conversion. The Model II is unusually quick with hexadecimal. Just enter &H before the hexadecimal number if you want the decimal in return.

I also use the OASIS operating system from Phase One Systems. That Basic has statements for conversion. HEX(string-expression) is hexadecimal to decimal conversion. HEXOF$(number-expression) is decimal to hexadecimal conversion.

The Sep/Oct issue had some NOTES on PEEK and POKE for Model II. I followed the instructions, but only the PEEK seems to work. I can't get the POKE to work. I need some help...

Donald M Dealy
EDP Director
S Attleboro, MA

(The PEEK/POKE routine in the Sep/Oct 80 issue, page 38, had one error. The first FIND string in the first patch should start with AF rather than C5. This correction was noted in the following issue in Corrections, and also in the issue following that, as a reminder in an answer to a letter to the editor. Keep in mind that this patch works only with TRSDOS 1.2 for the Model II. Does anyone know how to do it to version 2.0? Ed)

I take exception to Cameron Brown's review of the Radio Shack "Versafile" software in 80-U.S. for Sep/Oct 80. The program has its attractions, perhaps, but its implementation is sadly lacking. Consideration for an unskilled user appears to be nil. Specifically:

1. The program does not AUTO into Basic, never mind all the way to RUN.
2. Disk drives have to be specified - the system should default to drive 0.
3. Operation requires the use of LIST and EDIT, neither of which is desirable. This also opens up the possibility of damage to the program, or the files, by unfamiliar users.
4. Jargon is rampant: "Global Search", "Multiple Kill", etc. This is not the sort of thing to reassure my mother-in-law.
5. On start-up, the user is faced with a blank screen (with a prompt symbol). Why no explanatory text?
6. On entering data or pressing ENTER, nothing happens until the prompt reappears. Why no reassurance message or confirmation display?
7. "Shall I continue (Y/N)" is not INKEYed and thus requires unnecessary key strokes.

The list could be continued. On the plus side the listed program is found to be well remarked, and it is thus easy to make the necessary changes. But should we have to, in a product that is offered for sale as a finished product?

M Barlow
Pierrefonds, Quebec

(The degree of sophistication in software and users must be climbing at an alarming rate. I must agree with most of your observations, but also remember when I first saw the program that I thought it was the neatest thing since zippers and sex. Ed)

This is probably the 1100th letter on this subject, but one never knows. Regarding program AEXIT, which was part of the article "Add New Commands to Level II" Sep/Oct 80 issue, the ORG statement needs to be 7F17 instead of 7F18 in line 330. The memory size is OK.

The problem as it stands can be seen in the assembled part of line 1820. It indicates 18H goes into 7FFFH, which means the E0H goes into the following byte (8000H). This byte does not respond in a 16K machine. The result is a lockup as the processor attempts to load the byte from 8000H.

The article was excellent. The programs seemed to be an afterthought, and were not well explained. The System/Command article in that issue wins my vote for the best article in any computer magazine, ever. (Howzzat for unbridled praise, Phil?) More, more.

Dan Connors
Keeper of the Penquin
Harvey, LA

Last night I read your Nov/Dec 80 issue. This morning, my Model II screen suddenly went dead with a squeal. I slapped the power switch. I'm happily on the air today. If there's any truth to your admonition on page 90, you've saved me considerable grief. It was certainly timely.

One very appreciative subscriber,

Gerald Lippey
Los Angeles, CA

(We are wondering if they fixed that in the 2.0 release of TRSDOS for the Model II, but have been too chicken to try it! Ed)

Fastload

FOR TRS-80* MODEL I USERS ONLY

16 Times Normal Speed

*TRS-80 is a trademark of Tandy Corp.

- High speed load TRS-80* Level II cassettes
- Input 15K byte Level II program in 15 seconds
- Search BASIC or SYSTEM programs by name

Unlike other high speed tape input devices, FASTLOAD uses standard format cassettes. Therefore, there is no need to re-record on other media. At 8000 baud, FASTLOAD is faster than disk for short programs. FASTLOAD reads tapes at the fast-forward speed of the CTR-41 cassette recorder. The recorder can also be used for CSAVE at the normal speed.

FASTLOAD connects to the 40 pin I/O or to the Expansion box. The control program does not use computer memory because it is in a built-in PROM. Other valuable features are keyboard debounce program, automatic key repeat routine and key-beep via cassette speaker. Price is $188.00 for FASTLOAD and $95.00 for the modified CTR-41 recorder.

Personal Micro Computers Inc.

475 Ellis Street, Mountain View, CA 94043 **(415) 962-0220**

80-U.S. INTERVIEWS SCOTT ADAMS

of ADVENTURE INTERNATIONAL

This issue we talk with Scott Adams of Adventure International. Scott is one of the pioneers in the microcomputer software business. He introduced high quality game software when many were coming out with trivial games or more versions of Star Trek.

In a time when people were getting taken to the cleaners by software authors whose programs were of questionable quality, Scott Adams' Adventure series gained widespread and well deserved reputation as high quality, enjoyable game software.

It's amazing to think that he did what he did with no more than a Level II 16K machine at the start. But he did it and we are all the richer for it.

Scott started playing Adventure in 1978 on the DEC-10 version of the game written by Crowther and Woods in FORTRAN. He took ten days to master the game and became a full fledged addict.

A Systems Programmer by training and experience, he brought to the TRS-80 a background suited for the introduction of his Adventure Interpreter, which made possible the development of his Adventure Series.

As with many new TRS-80 owners, his wife Alexis at first opposed his new love. She hid his disks and almost ended Adventure before it began when she chose the kitchen oven as a hiding place. Later on, Alexis not only relented, but collaborated on one Adventure and wrote another herself.

Scott is an avid Science Fiction buff. He has over 3000 titles in his collection and admits to always having wanted to be a Science Fiction writer. This gives him a unique perspective which has enriched the world of TRS-80.

80-U.S. - You used to be a Systems Programmer for Stromberg-Carlson?
Scott - My last gainful employment.
80-U.S. - Not everyone would agree that you are not gainfully employed now. Let me just go over the information I have to make sure it's all straight. You learned about Adventure by playing the DEC-10 version of the game?
Scott - Right.

80-U.S. - According to your article in December's Byte magazine, the big thing that you gave to Adventure was the idea of the Adventure Interpreter. To your knowledge, has anyone else come up with the same thing?
Scott - As far as I know, no one to date.
80-U.S. - What gave you the idea for the interpreter?

Scott - It just seemed the way to do it.
80-U.S. - I suppose a Systems Programmer would take to that naturally.
Scott - Yes. I worked on compilers and interpreters at Stromberg...and operating systems. It's a very structured way of thinking when you are going to do a task - you do it so that it will be re-usable.
80-U.S. - It makes sense. It's strange that no one else came up with the idea before.
Scott - Well, one of the problems with an interpreter is that it's a *bear* to design. Basically, I had to decide what the output looks like, what the input looks like, and how to combine something internally that understands it. It's easy to do a straight forward Adventure, say in Basic, where you are reacting to something the player says. It's just loops and IF statements. An interpreter doesn't do anything. It doesn't respond to anything. It is just a language. Then you have to write the Adventure, so it's a two step process.

80-U.S. - But once the interpreter is done, I expect that it makes it considerably easier to write follow-on Adventures.
Scott - Right. Once you're used to writing in it. I developed it of course, so I am used to writing in it.

80-U.S. - I noticed that you and your wife collaborated on the adventure you published in the December issue of Byte - the Pirate's Adventure. Was that mostly her idea or yours?
Scott - It was sort of like, I'd say: "OK, what do you want to do now?" And she would say, "Well, let's see. Let's have a cave." "OK, now what are we going to do with a cave? Where can we put it so it will be logical?" We'd go through like that. The fourth adventure, which is Voodoo Castle, she did almost 95% on her own.

80-U.S. - Did she!
Scott - I think she just wanted to see if she could do one to get it out of her system. She hasn't tried one since.
80-U.S. - Has she any background in programming?

YOUR MODEL II CAN HAVE SNAPP!

SNAPP II EXTENDED BASIC

A family of enhancements to the Model II BASIC interpreter. Part of the package originated with the best of APPARAT, INC.'s thoughts in implementing NEWDOS BASIC. The system is written entirely in machine language for SUPER FAST execution. The extensions are fully integrated into Model II BASIC and require NO user memory, and NO user disk space. The package is made up of the following six modules, each of which may be purchased separately.

XDASIC—Six single key stroke commands to list the first, last, previous, next, or current program line, or to edit the current line. Includes quick way to recover BASIC program following a NEW or system or accidental re-boot. Ten single character abbreviations for frequently used commands: AUTO, CLS, DELETE, EDIT, KILL, LIST, MERGE, NEW, LLIST, and SYSTEM. $40

XREF—A powerful cross-reference facility with output to display and/or printer. Trace a variable through the code. Determine easily if a variable is in use. $40

XDUMP—Permits the programmer to display and/or print the value of any or all program variables. Identifies the variable type for all variables. Each element of any array is listed separately. $40

XRENUM—An enhanced program line renumbering facility which allows specification of an upper limit of the block of lines to be renumbered, supports relocation of renumbered blocks of code, and supports duplication of blocks of code. $40

XFIND—A cross reference facility for key words and character strings, also includes global replacement of keywards. $40

XCOMPRESS—Compress your BASIC programs to an absolute minimum. Removes extraneous information; merge lines; even deletes statements which could not be executed. Typically saves 30-40% space even for programs without REM statements! Also results in 7-10% improvement in execution speed. $40

ENTIRE PACKAGE ONLY $200.00

DOSFIX

A collection of patches to TRSDOS and BASIC to enhance their usability and function includes our well-known BREAK7E patches to keep the break key from being used accidentally. FREE WITH ANY MODEL II SOFTWARE PACKAGE.

CONVERT

This remarkable utility converts 'V' format files (the sequential format used by the SHACKS, COBAL and BASIC Compilers) to the 'F' format files (the sequential file format used by the BASIC interpreter and DASCOM), and vice versa. Without this product programs written for the interpreter will have to be RE-KEYED to be used by the SHACKS Compiler BASIC. $75

SBASIC — Model I and Model II

Program in a high-level, full structured BASIC! The BEST of the BASIC pre-processors. PERFORM named subroutines, CONDITIONAL case structures, WHILE loops, UNTIL loops. And much more. Forget about line numbers. Model II version is compiled and SUPER FAST. From Ultimate Computer Systems, Model I $50 Model II $75

XPRINT

Print neatly formated hard copy listings of BASIC programs from disk. Programs may be ASCII or compressed. Quick and easy group selection allows you to print many listings with one command. $35

BPRINT

Allows you to access a serial printer simultaneously with the standard parallel printer. Easy interface to BASIC. Drive two printers at once! $75

PPRINT

Updates to The Electric Pencil to support true proportionally spaced printing with the Shacks new letter quality printer, the Daisy Wheel II. Produces copy which looks as if it had been typeset. $100

ITOII

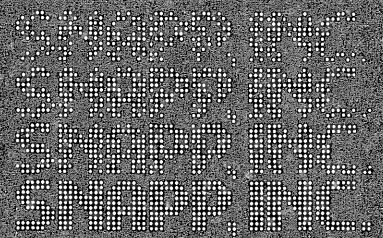
A helping hand when converting BASIC programs from the Model I to the Model II. Automatically adjusts PRINT @, and PRINT USING to compensate for differences in the language. Advises you where adjustments are necessary for PEEK, POKE, etc. $25

EXTENDED BUILT IN FUNCTIONS

Now you can give your TRS-80 all the functions you wished BASIC had given you in the first place. These verbs will give you programming abilities that make you look good. Adds the following function verbs: SORT, PEEK, PEEKW, POKE, POKEW, ETIM$ and XTIM$. $50

DIAL

USR 330D Auto Answer/Auto Dial, Direct Connect Modem, 300 baud, originates/answers 103J compatible. When used in conjunction with our DIAL, software is capable of complete origination of communications with remote locations without operator intervention. Special combinatian price, modem and software. $430
Software only $50



8160 Corporate Park Dr.
Cincinnati, Ohio 45242

Ohio residents call collect
(513) 891-4496

Call Toll Free
1 - 800 - 543-4628

All products now available to run with TRSDOS 2.0.

Most products will soon be available for the Model III. CALL FOR DETAILS!

MASTER / SLAVE

This software package was designed to support the transferring of files from one Model II to another via direct connection or modem/phone line connection. ALL kinds of files and baud rates up to 9600 are fully supported. Transfer files in either direction, even with the SLAVE Model II UNATTENDED! $150

SPOOLER—Model I and Model II

Our workhorse! Unlike the one supplied with TRSDOS 2.0, ours requires no special knowledge or training on the part of the operator. Additionally ours performs much better. On the Tandy SPOOLER, everytime a disk is accessed, the printer stops dead! This package is available for Model I, in the TRSDOS/NEWDOS 80 versions, or for the Model II. Greatly enhances system performance when running typical business applications. Many applications have been benchmarked to run nearly TWICE AS FAST with the SPOOLER installed. Installs in minutes and no changes are required to your programs. Preferred Model II versions require NO user memory. Optional features for the Model II version only:

Serial printer support, and DISK SPOOLING support is particularly recommended for word processing applications $100
SERIAL PRINTER OPTION $50
DISK SPOOLING OPTION $50

HOSTII/TERMII

Allows remote control of a Model II from another Model II or any ASCII terminal. Our Host system, unlike the one supplied with TRSDOS 2.0, supports accurate screen positioning on the Term station. Without this feature, formated displays appear on the terminal looking like randomly placed garbage. Requires NO user memory! This system is designed to provide software support to our customer locations without ever leaving the office. $50

TERMS OF SALE:

Credit card customers, add 3%. C.O.D. customers add $3. Ohio residents add 4½% sales tax. Shipments normally made the same day we receive your order.

OUR GUARANTEE:

If your diskette arrives damaged, we will replace it without charge. If you ever accidentally damage it, we will replace it for a $10 handling charge. For a period of one year, we will provide you with any enhancements or updates for a $10 handling charge. For a period of one year, if errors are discovered in the programs they will be corrected without charge. In the event we cannot correct an error, you may return the program material for a refund.

Electric Pencil is a trade mark of Michael Schrayer Software, Inc.

TRS-80 is a trademark of the Radio Shack division of Tandy Corporation.

NEWDOS and NEWDOS/80 are trademarks of Apparat, Inc.

Circle 7

Scott - No.
80-U.S. - That must make your interpreter very easy to work with.
Scott - Well, it doesn't really require the ability to think like a programmer. It's not really a programming language. It is more of a table driven language. It's hard to describe. I've had other people work with it. Adventure 8 was a collaboration. We worked on that on and off for about a year before it was released. For some people it is hard to pick up a language and also the philosophy of how to play Adventure.
80-U.S. - But it certainly made it possible to get something reasonable on a small system.
Scott - It used to take me... oh ... about an hour to do...I guess you'd call it "compile" an Adventure, because I was running the interpreter with about 16K plus the editor language on the 16K machine. It would have to write a data base which would take about 20 minutes to write to tape and then it would take another 20 minutes for the interpreter to read it back in.
80-U.S. - That's how computing was done back in those days. Amazingly enough, that's not that far back in time.
Scott - If you think about it, it seems amazing that people still talk to large million-dollar machines using cards with holes in them.
80-U.S. - It surely doesn't make the time you took on your Level II 16K sound very bad.
Scott - No it doesn't. The worst problem was coming back after a 20 minute load and finding you have an error just before the end!
80-U.S. - Are you planning on extending any of your Adventures to some of the other TRS'80's like the Model II or III or Color Computer?
Scott - I've already got nine adventures for the Model II. The entire series of ten work on the Model III. We have all three, Model I, II, and III working. The color computer...probably, sometime in 1981. I don't know when for sure. Whenever we can work on it. We are also putting the games on the ATARI. Another project is the CP/M system. It will probably be the Z80 system since the code is already in Z80 and I don't want to back it down to 8080. Most people are getting a Z80 board one way or another.
80-U.S. - If nothing else too, you can use your Model II as a good CP/M development system for that.
Scott - That's a possibility. I do have a Model II CP/M, but I haven't really cracked it. It's still in the bag. There are always more projects than there is time for.
80-U.S. - Yes, everybody seems to be in that situation.
Scott - We are also in publishing now. We're in the top 5 of the largest publishing houses, and have got...oh..140 to 200 items right now.
80-U.S. - What made you decide to get into marketing these games?
Scott - I took them to club meetings and showed them to people. At that time I submitted them to Sensational Software and the Software Exchange. In the long run they both dropped the ball, in terms of really pushing it. So I decided to do it on my own. What eventually evolved was the publishing company. It just took off. There are a lot of software houses out there that have good products they want to publish like I did. I guess it's been over a year since I was at Stromberg, and it's going well.

80-U.S. - Have you ever had any experience with running a business before?
Scott - Not really. That was my minor in school, with Computer Science. My wife ran a couple of restaurants. It's a lot of work, but fun too. I have a place to publish my software such as Adventure International. I can look at some of the monthly reports to see what my royalties would be now if I paid myself. They would be very, very nice. I've got some authors who are getting over a thousand a month.
80-U.S. - How do you go about selecting software for your line?
Scott - People send it in and it's reviewed. Five areas are evaluated. It must be well written. That's really as important as whether it is saleable. A program a person will shell out X dollars for should give them their money's worth. That's a very subjective opinion. We have done pretty good. Arcade games are really popular. We also tend toward complex simulations. That's what Adventure really is all about. We just came out with our first major business package, which is Maxi Manager. It has been reviewed as one of the better data management packages around.

**'80-U.S.—What percent of submittals do you actually accept?
Scott - About 5%.'**

80-U.S. - Yes, I've just been looking at an ad for Maxi Manager. Has it been reviewed recently?
Scott - I don't think it has gotten into any magazines. The reviews came from people who have used it, people who have used a number of data managers and like it.
80-U.S. - Do you plan to be expanding your line in business and other software?
Scott - That depends on what's submitted to us. We are a free-lance publishing business, so it depends.
80-U.S. - I see.
Scott - We have a small in-house programming effort, but a lot of software is submitted to us.
80-U.S. - I can imagine. What percent of submittals do you actually accept?
Scott - About 5%.
80-U.S. - How about getting into more realistic simulations rather than fantasy simulations?
Scott - It depends on the submissions. We've stayed out of the pure educational field. A lot of our stuff can be viewed as educational, but it's only incidental. It is primarily designed to be fun. Project Omega is a real life simulation, but it is probably 25 years ahead of it's time, building the first L5 space colony. In that respect it is not fantasy. It's all hard science and well researched. But once again, it's still fun.
80-U.S. - Right.
Scott - It's more science fiction than say, science fantasy.
80-U.S. - Are there any more adventures on the way?
Scott - Number 10 has been released. It's called Savage Island, Part 1. There will be at least one if not more follow-ups. It is the first multi-part adventure. There is also the project I've been working on for the last six months, which is rewriting the Adventure Interpreter. The original interpreter was optimized for 16K, the new language is going to be for 128K.
80-U.S. - It sounds like it will be rather interesting when it gets done.
Scott - One problem is that publishing has pulled me away from the writing. I'm at the point now where I'm going to hire an Assistant Publisher. By the time this is in print, we will already have him. He will share some of the duties so I have some time to go back and do more writing.
80-U.S. - How big an organization is Adventure International?
Scott - About 15 people. I'm not sure, I don't do the payroll.
80-U.S. - Mostly concerned with keeping track of things?
Scott - They are involved in manufacturing, advertising, clerical and management.
80-U.S. - Are you the only programmer among them?
Scott - Well, right now, but that will be changed soon.
80-U.S. - I can believe that would be necessary. Do you have anything you would like to suggest to anyone who would like to submit software to you to think about before they go ahead and put it in the mail?
Scott - Usually the biggest problem is that things are not completely thought out. The greatest waste of time is sending in things that would at best make a good short magazine article. Like writing an alarm clock using the computer to wake you up in the morning. Also, you have to keep in mind the package has to give the user full value for his money. We usually judge the package by how much time you would spend on the package and compare it with a show. If you would shell out three dollars for a two hour movie, you would like at least a dollar fifty an hour return on your investment if you stay home to play software. So if you charge 15 bucks for a package, you would expect to get at least 6 hours out it.
80-U.S. - From talking to people, they feel they get that and more from your software.
Scott - I hope so. ●

Now Inexpensive Word Processing with
Starwriter Daisy Wheel Printer

$1795

Starwriter/Starwriter II Daisy-Wheel Printer

With a print speed of 25 cps, the Starwriter incorporates the latest LSI technology and employs an industry-standard 96-character print wheel--an easy-to-change format that readily accommodates muti-lingual applications. Starwriters produce letter-quality printing on three sharp copies with either 136 columns or 163 columns. Self-test capability, programmable VFU (vertical format unit) and front panel indicator lamps for paper, select and power status. Compatible with sheet-feeders and can handle up to 15 inches wide. Tractor optional.

- ★Bi-Directional
- ★Logic seeking
- ★136 printable columns
- ★25 CPS printspeed
- ★Elite pitch
- ★Pica pitch
- ★40 msec. line feed time
- ★Uses standard Diablo ribbon
- ★1/48 inch minimum line spacing
- ★Standard parallel or RS232-C* compatible
- ★115V ± 10%, 50/60 Hz, 70 W power requirements

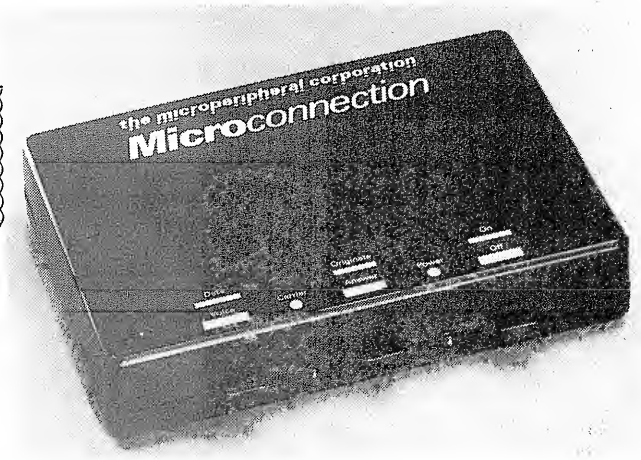
*Add $60.00 for Serial

Ask about Starwriter II 45 cps

LORDS
Small Systems Design
P.O. Box 99
Port Angeles, WA 98362
Telephone (206) 457-3064

Circle 8

New Products



Music Sweetener

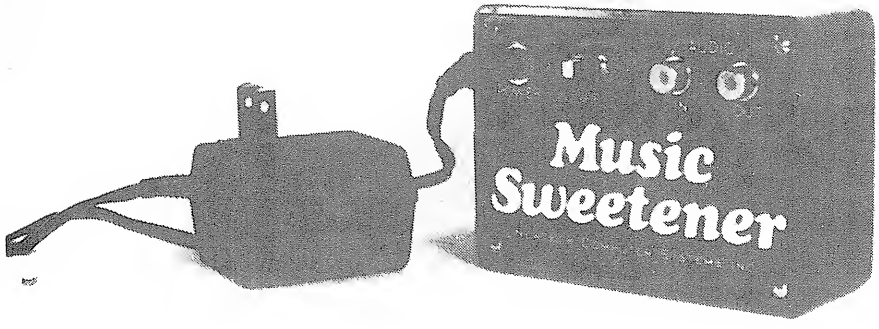
The Music Sweetener is a low-pass filter designed to improve the sound quality of Software Affair's Orchestra-80 and other commercial and homebrew digital-to-analog converter music synthesizers that do not already incorporate a filter into their design. The Music Sweetener attenuates the unwanted high frequency sampling noise four times better than a stereo system's treble control, and results in a much more distortion free, and pleasing sound. The Music Sweetener is similar to the four-pole 3.2Khz low-pass filter incorporated into Newtech's Music Box for the Model I TRS-80, and is optimized for four-part music synthesis on most micro-computers. The Music Sweetener is simply inserted between your music peripheral and your audio amplifier. The Music Sweetener complete with instructions and AC adaptor power supply is $39.95 plus $2.00 for shipping and handling from Newtech Computer Systems Inc., 230 Clinton Street, Brooklyn, NY 11201 (212) 625-6220

Basic Compiler

Microsoft Consumer Products has just released a new version of the Model I TRS-80 Basic Compiler. Owners of the Basic Compiler who have a signed non-disclosure statement on file, will receive the upgraded Compiler. Owners who have not submitted a non-disclosure statement, should contact Microsoft Consumer Products, 10800 N E 8th Suite 507, Bellevue, WA 98004

Income Tax Programs

The 1980 Gooth Income Tax book is now ready for mailing. This 128 page book makes microcomputers helpful in doing one's own taxes, or doing taxes for others. More than 40 Income Tax Form programs are included with line by line listings. Also included are programs for rental statements, installment sale statements, earned income credit and a program that figures the taxable portion of unemployment pay. The programs are written in TRS-80 Model I Basic with several pages explaining the TRS-80's commands for conversion to other Basics. Also, the chapters on depreciation of microcomputers, and programming tips for newcomers should prove helpful. Priced at $16.95 postpaid, Visa/MC accepted. Gooth Software, 931 South Bemiston, St Louis, MO 63105



Microconnection

The Micro-Peripheral Corporation has just introduced an exciting new line of direct connect modems. The products, called "Microconnections", are designed to interface most popular computers and terminals to the telephone network. All units feature Bell 103 compatible operation in the originate or answer mode. A direct connection to the telephone line eliminates the problems associated with acoustic coupled modems and provides high sensitivity, low error rates and noise free operation. The TRS-80 Connection interfaces directly to the Model I and PMC-80 data/address bus to decode RS 232 information. It can be connected directly to the keyboard, which eliminates the need for the expansion interface or serial I/O RS 232 board. The RS 232 Connection plugs into the standard DB25 connector. It provides an ideal replacement for obsolete acoustic coupled devices and can be used with virtually any computer or terminal with provision for RS 232 serial I/O at 300 baud.

There is provision for adding an AutoDial/AutoAnswer module to all Microconnections. This device permits automatic connection to other computers via the telephone network, with unattended data transfer such as message sending and retrieval. The AutoAnswer provision permits unattended remote access of the host computer. A detector is provided for ring counting and preset answer conditions.

Another option allows the Micro-connection to be used with European systems (the European Connection). Tone frequencies are set to CCIR standards. Microconnection prices start at $199.95. The AutoDial/AutoAnswer option is $79.95 while provision for European standards adds $29.95. MicroPeripheral Corporation, 2643 151st Place N E., Redmond, WA 98052 (206) 881-7544

Conquest

Lakefront Software is proud to introduce its newest addition, a strategic simulation entitled Conquest. This game is for two players and is based on the famous board game of world conquest. The play is set in outer space, with two factions, you and your opponent, both fighting for control of the newly discovered planet. This is not an acrade gamel You must deploy armies, fight off invasion of your colonies and plan your attacks carefully. A game may take from one to three hours to play depending on your strategies. Conquest is available on cassette for TRS-80 Level II 16K. It is priced at $14.95 from Lakefront Software, PO Box 5240, Willowick, OH 44094.

New Business Programs

Management Systems Software Inc. introduces three new programs for the TRS-80 user. The first program, called PROCURE, ascertains the optimal amount of a commodity to purchase if the purchase price varies over time. This program takes into account the inventory capacity, beginning inventory levels, and anticipated useage in ascertaining the optimal procurement amount. Price is $150.

The second program is a series of 13 business programs including Ratio Analysis, Sources and Uses, Optimal Cash Balance, A/R, Economic Order Quantity, Debt/Equity, IRR and NPV, Depreciation Schedule, Optimal Tax Life, Cost of Capital, Sensitivity Analysis, Capital Budgeting Analysis and Risk-Adjusted Discount Rate. Price is $200.

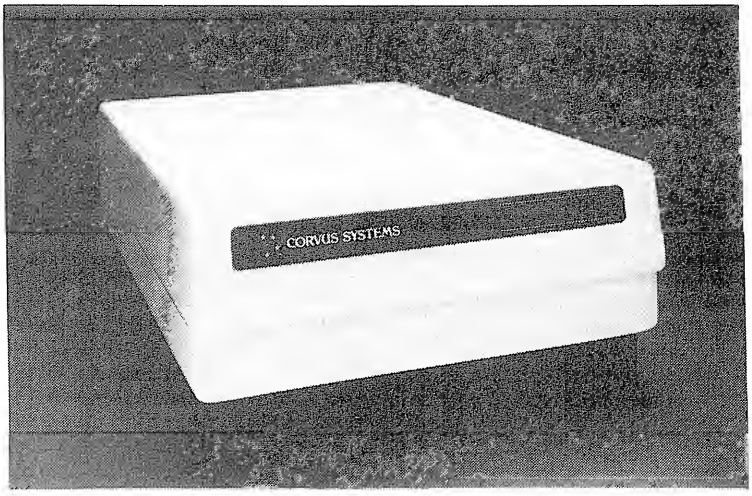
The third program is an enrollment projection program. This program will forecast, for up to five years in the future, university, college and community college enrollment. Price is $100. Management Systems Software Inc., 5200 Brittany Drive #1006, St Petersburg, FL 33715 (813) 864-4347

Computerized Index

Hexagon Systems has released a computerized index to a number of popular microcomputer magazines, Kilobaud, 80-Microcomputing and 80-U.S. Journal. Nearly 700 items of interest to the owner of a TRS-80 are included. Items indexed include technical tips, programs, reviews and advertising.

Powerful search and scan programs are used to locate items in the index. The "Scan" program can search through index text to locate a keyword, or a series of keywords linked by AND or OR. The "Search" program searches by item description. Complex inquiries such as "Accounting or Inventory" AND "NOT HOME" AND "80-U.S. OR KILOBAUD" can be built, then hundreds of items searched in less than a second.

A 48K TRS-80 Model I with 2 disks is required. The programs, index and manual are available for $29 from Hexagon Systems, PO Box 397 Station A, Vancouver, BC Canada V6C 2N2. The manual is obtainable separately for $8.



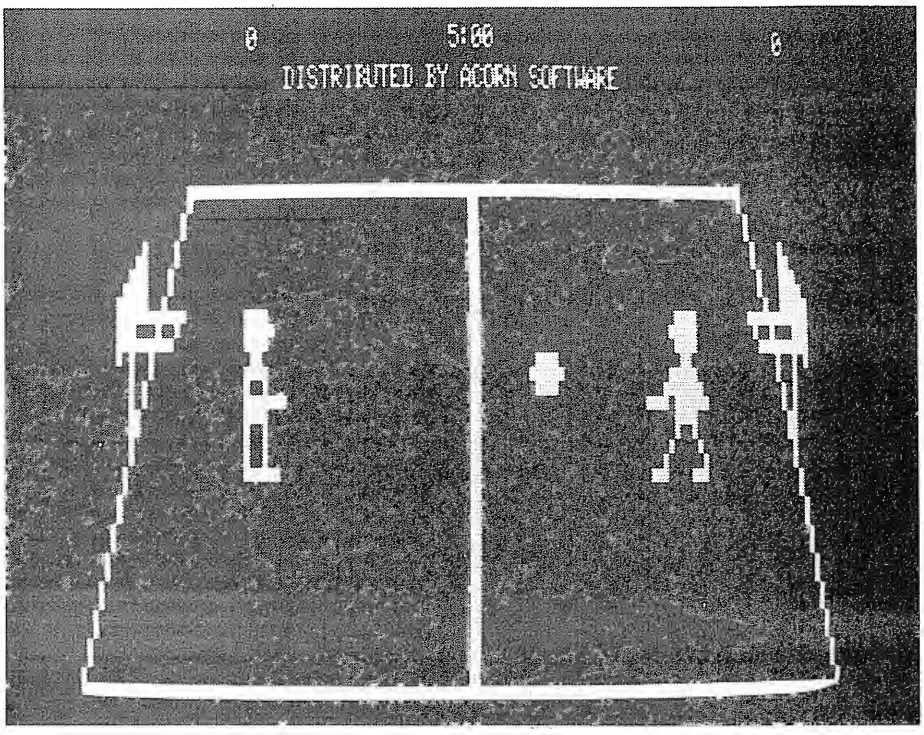
New 20 MByte Disk

The newest addition to the growing line of intelligent peripherals from Corvus is a new 20 megabyte version of their highly successful Winchester disk system. While providing twice the capacity of the Corvus 10 megabyte disk, the new 20 megabyte system sells for only 20% more. Like the 10 megabyte model (which continues in the product line), the 20 megabyte disk is fully compatible in both hardware and software with some 15 of the most popular microcomputers including TRS-80 Model I and II. For applications requiring greater than 20 megabytes, up to three add on disks with either 10 or 20 megabyte capacity can be incorporated into the system. Price of the 20 megabyte system is $6450 including Winchester drive, Corvus controller and personality module. Add-on disk drives are priced at $5750. Delivery is 30 to 45 days after receipt of order. Corvus Systems, Inc., 2029 O'Toole Ave., San Jose, CA 95131 (408) 946-7700

Basketball

Acorn Software Products, Inc. announces the debut of Basketball, a new action-packed, one-on-one basketball game program by John Allen.

Basketball is designed for the Model I Level II TRS-80. The program features a five-key player control range allowing defensive play backwards, defensive coverage and offensive maneuvers from side to side, offensive drives forward down court, and shooting/scoring capability at any time beyond the center line. Single player option offers five levels of difficulty. Basketball is priced at $14.95 on cassette or $20.95 on diskette. Acorn Software Products, Inc., 634 North Carolina Ave., SE, Washington, DC 20003 (202)544-4259



ITEMS AT RANDOM

New Managing Editor

Tom Huber has joined the staff at 80-U.S. He currently is looking over everyone's shoulder, and just as soon as we can knock a hole in the wall to make more space, he will become our Managing Editor. We all wish him well and hope his stay at 80-U.S. will be enjoyable and rewarding.

80-U.S. goes Reader Service

With this issue, we are installing a full reader service. We couldn't get the "Circle #" into every ad, so check the advertiser index on page 120 if you are in doubt. Circle the numbers for which you want more information and send in the card. You should be hearing directly from the advertiser after that.

New Foreign Subscription Rates

Because the Post Office has made a major jump in rates, we must increase our rates to foreign subscribers. All foreign subscriptions are now for one year only, and all foreign prices are $30.00 per year, except for Canada, which is $20.00.

Attention Canadian Subscribers!

We must apologize for getting your Jan/Feb 81 issue out so late. Our Canadian distributor decided to quit, and sent us your names and addresses in January. Starting with this issue, you will be handled through 80-U.S. in Tacoma again, and your copies will be mailed via 2nd Class, along with the U.S. subscribers.

Things we would like to see

How about a Model III to Model I bus? Something that would allow you to use your Model III with existing Model I attachments to the bus, like the parallel printer cable, voice synthesizer, etc?

How about programs on Astronomy that could give the position of any star for any time?

How do you use a computer in your business? Is there some novel or unique way you use it? What you take for granted may provide enlightenment for someone else. Same holds true for engineering, education and medicine. We found that some of the most useful programs around here are short, sweet and generated in a hurry to do a specific job on the spur of the moment. How about yours?

New Radio Shack Line Printer

It has come to our attention that the Line Printer III is being replaced by the Line Printer V, a faster version of the III. It runs at 160 characters per second, and supports some graphics. It will be in the same package as the LP III. Tandy lowered its prices for the printer and ribbons (a welcome change of price direction). The ribbon price is $13.95, down from $21.95; the printer is $1860, down from $1960.

Copy Machine?

Word has it that the Radio Shack Computer Centers will start carrying copiers soon (bond copiers?). The first is supposed to sell for less than $800. When are they going to make a bond copier that will act as a printer for your computer?

In this Issue

Not just another Tic-Tac-Toe game, Steve Kelley has put together a real artificial intelligence program here. It doesn't have all the possibilities and eliminate losers, it *learns* the right moves from the start. Yes, it's pretty dumb at first, but let it play itself for about 2000 games, then watch out!

Our feature this issue is VisiCalc, a program that has been around for a while for the Apple and others. This program is so full of possibilities it staggers the imagination. We have used it for several projections here and it's good!

Spencer Hall gives us another program to figure out your budget, with a twist. It can tell you who to short so that you can keep your allowance. Not bad Spence...

Bill Schroeder takes a hard look at TRSDOS 2.0 for the Model II. It's good, but still not quite there according to Bill.

David Busch comes through twice in this issue. First with his article on gaining copy flexibility with NEWDOS/80 and again with his six ways to get more out of Scripsit.

Bill Barden Jr starts a mini series on the Color Computer in this issue. It looks like we are going to see a lot of this new computer, and we look forward to more from Bill.

J C Dahlke submits a program to figure water flow over a dam. *No, this isn't an April fool!* The program is for the TRS-80 16K Level II and up, and shows an interesting engineering application.

Go ahead, read the rest for yourself, there's no more space here. It's all somewhere between here and page 120, hope you enjoy it!

Corrections

In our Jan/Feb 81 issue on page 74, we failed to mention that the program Slalom Run, by Greg Perry, will also run on the Model III, 16K Level II and up. It's a little faster though, and you have to dodge the trees more expertly than on Model I.

Spencer Hall's Nine Z-Subs on page 26 of that issue drew much comment. There is a blatant error in his article, in that in the center column, item 7 appears twice. The first occurance of number 7 should be ignored.

Page 51 of the Jan/Feb 81 issue carried an ad for LDOS. The first item on that ad indicated that you had to "ZAP this and that" to get it to work. Not so, our typesetter missed a word, and it should have read "without ZAPPING THIS and PATCHING THAT". We must have been forgiven, since LDOS appears in color in this issue on pages 16 and 17.

Another boo-boo, although not serious since it was in a remark, was in the listing for System/Command of Jan/Feb 81, page 53, line 420. OKAY to SNED, should be OKAY TO SEND.

On page 115 of that issue we apparently put our issue to *be* instead of *bed*. Second line of the article on that page contains the culprit. Where *were* those proofreaders last issue??

We recently received a rather interesting order from New Zealand. Our order card was marked to indicate that the issues were A$, Air Mail was marked as B$ and the Visa was marked as being C$. The card then read: Please send A$ USING B$: CHARGE TO C$.

User Group

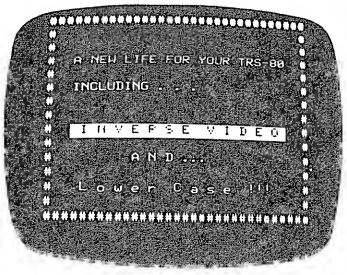
Psychologists who are interested in a user group of their own, built around the TRS-80, should contact Steven E Skindell, Macomb-Oakland Regional Center, 3377 Elizabeth Lake Road., Pontiac, MI 48054. He is calling the group PSY80 and will serve as an exchange of information on computers used in the field of mental health.

Hope you enjoy the fact that we finally have some color inside this mag. It's still not where we want it to be, but you gotta crawl before you can walk etc.

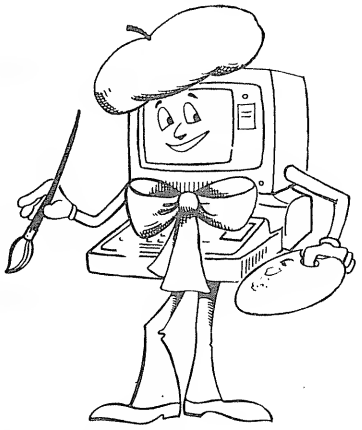
Pay your taxes, tell them you saw it in the JOURNAL, and remember that nice days (even days like April 15th) are made, not had...

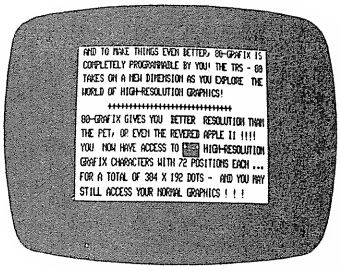
Mike

FROM PROGRAMMA
HI-RESOLUTION GRAPHICS FOR THE TRS-80®

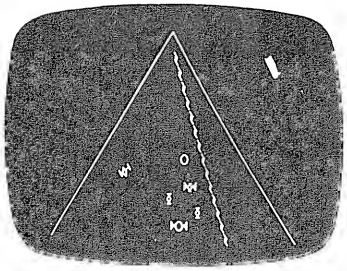


LOWER CASE
The 80-GRAFIX board includes two sets of lower case characters at no additional cost.





INVERSE VIDEO
The 80-GRAFIX board allows you to do inverse video to high-light your screen displays.



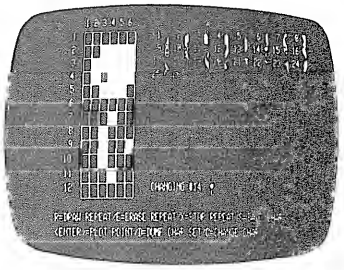
DEMONSTRATION PROGRAMS
The 80-GRAFIX board is supplied with a Character Generator software and several demonstration programs.

FINALLY, AT LAST. . .

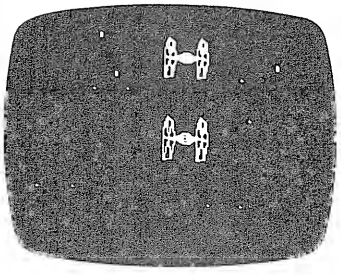
HI- RESOLUTION GRAPHICS is available for your TRS-80 computer system. The 80-GRAFIX board from PROGRAMMA International, Inc. gives your TRS-80 high resolution capability that is greater than the Commodore CBM/PET or even the revered APPLE II.

80-GRAFIX gives the TRS-80 an effective screen of 384X192 pixels, versus the normal 127X192 for the TRS-80, 80X50 for the CBM/PET, or the 280X192 of an APPLE II. As an added feature, 80-GRAFIX offers you lower case characters at no additional cost. Of course, you can also create your own set of up to 64 original characters using the supplied Character Generator software.

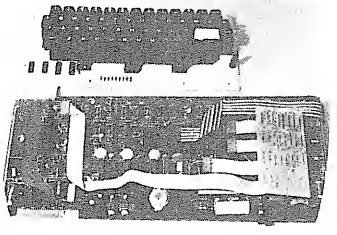
The 80-GRAFIX board is simple to install (note that this voids your Radio Shack warranty), and programming is done through BASIC. 80-GRAFIX opens up a whole new realm of software development and excitement never dreamed of for the TRS-80!



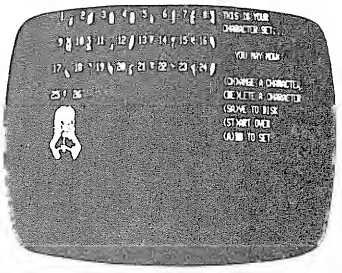
CHARACTER GENERATOR
The supplied character generator software allows you to create your own character set of up to 64 original characters.



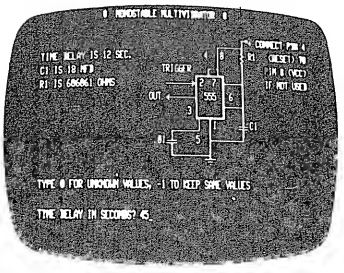
REAL-TIME GRAPHIC GAMES
With the 80-GRAFIX board you can write exciting real-time games using BASIC.



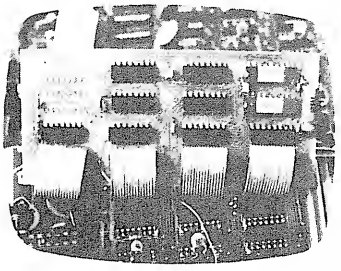
EASY INSTALLATION
The 80-GRAFIX board is simple to install and fits inside the TRS-80 case.



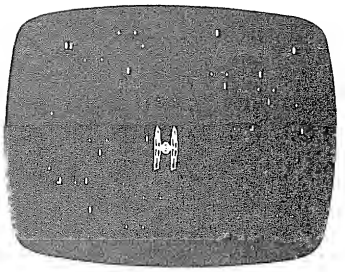
GRAPHICS GALORE
The 80-GRAFIX board and the supplied Character Generator allow you to become an artist.



ELECTRONIC DESIGN
The 80-GRAFIX board has unlimited application in Electronic design and Education.



80-GRAFIX HI-RESOLUTION
Finally, the only means to protect your computer investment is to order an 80-GRAFIX board TODAY!



EXCITEMENT & FUN
Open up a new realm of software development with the 80-GRAFIX board.

Available exclusively through PROGRAMMA at the cost of $149.95
Please check with us for availability prior to ordering
VISA and MASTERCHARGE accepted
TRS-80 is a registered trademark of the Tandy Corp.

PROGRAMMA INTERNATIONAL, INC. Circle 10
3400 Wilshire Blvd.
Los Angeles, CA 90010
(213) 384-0579 • 384-1116 • 384-1117

Expand your TRS-80*

With The LOBO LX-80 Expansion Interface

Now you can realize all the power and potential of your TRS-80*, Model 1. If it's add-on memory you need, your LX-80 can accommodate up to four 5¼-inch, single- or double-density 35, 40 or 80 track mini-floppies, four 8-inch floppies (single or double sided), and up to eight Winchester fixed disk-drives (5¼". 8", 14").

LOBO's powerful new LDOS™ operating system, provided with your LX-80, allows for the use of any eight drives, in any combination, single or double density.

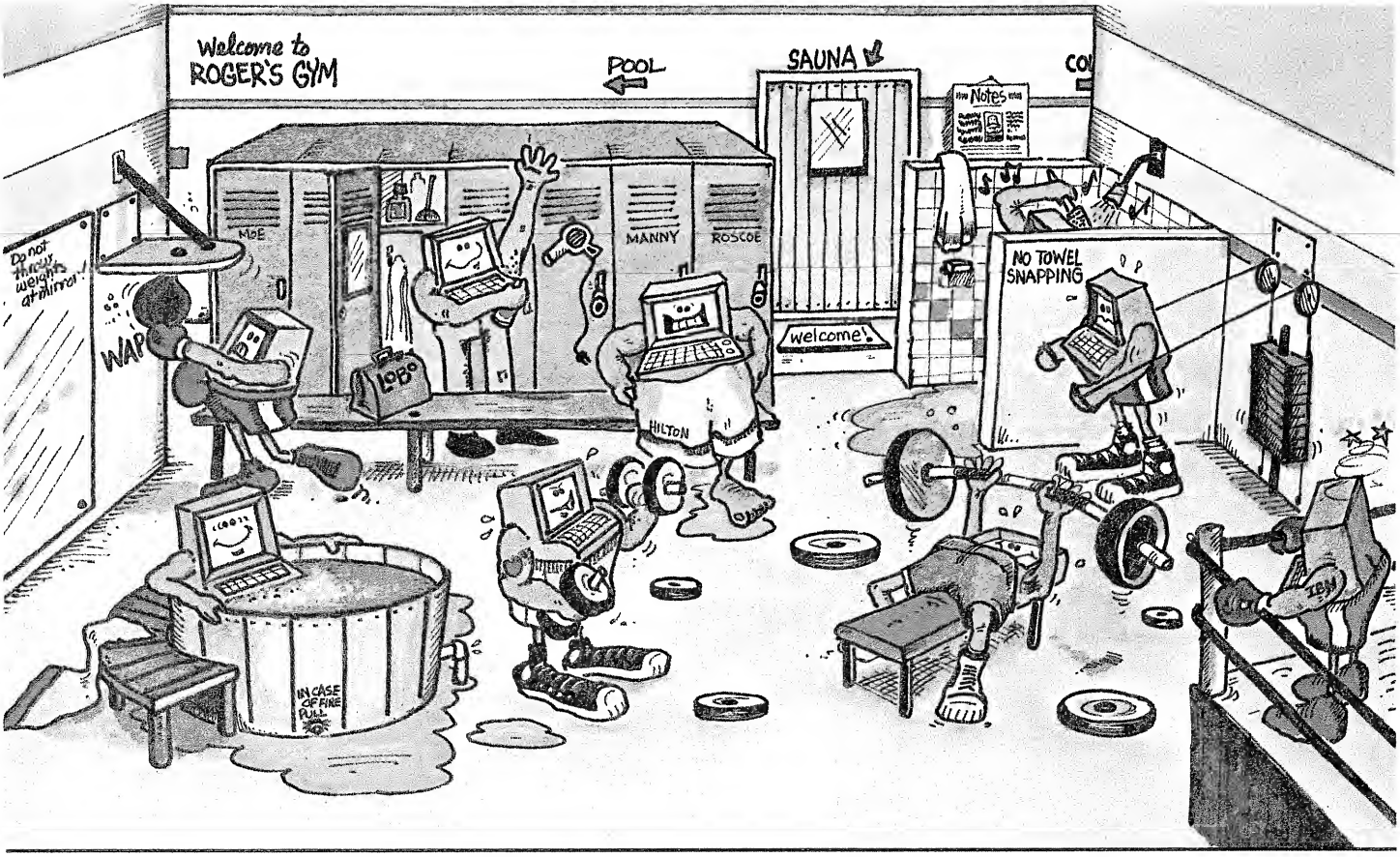
And there's more ... lots more. There are two parallel ports (standard), two serial ports (optional), a keyboard ROM override switch, and a 32K memory expansion (optional). Send for a free LX-80/TRS-80 cost performance comparison chart.

For the full story on how the LX-80 can expand your TRS-80, see your nearest LOBO dealer, or write or call:

*TRS-80 is a registered trademark of Radio Shack, a Tandy Company.

LOBO drives INTERNATIONAL

LOBO DRIVES INT'L
354 South Fairview Ave.
Goleta, CA 93117
(805) 683-1576



LOBO'S NEW LDOS™ Puts muscle in your TRS-80*

LOBO DRIVES' new LDOS™ Disk Operating System is loaded with outstanding features that will enable you to realize the full power and potential of your TRS-80.* With LDOS, you can support up to eight drives (5¼ and 8-inch drives, double-sided drives, double-density drives, 80-track drives), including the new 8-inch and 5¼-inch Winchester fixed disk drives, in any combination.

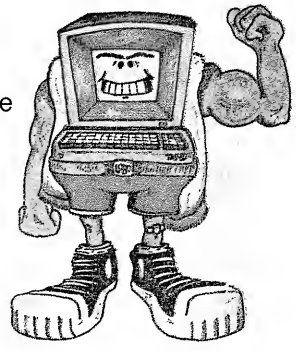
Other LDOS muscle building features include: ISAM accessing techniques; keyboard typeahead; Graphic string packer; Dated files, Marked files; File transfer by class; Built-in lower case display drivers; Non-breakable AUTO and DO commands, and many, many more.

LDOS is the perfect operating system to use with your LOBO DRIVES LX-50 or LX-80 expansion interface and disk drive subsystems. There's even an 800 number for instant service. To find out how you can put more muscle into your TRS-80, contact your nearest LOBO Drives dealer or call or write.

LDOS is available:

A. Operating System Diskette with Master Reference Manual
$139

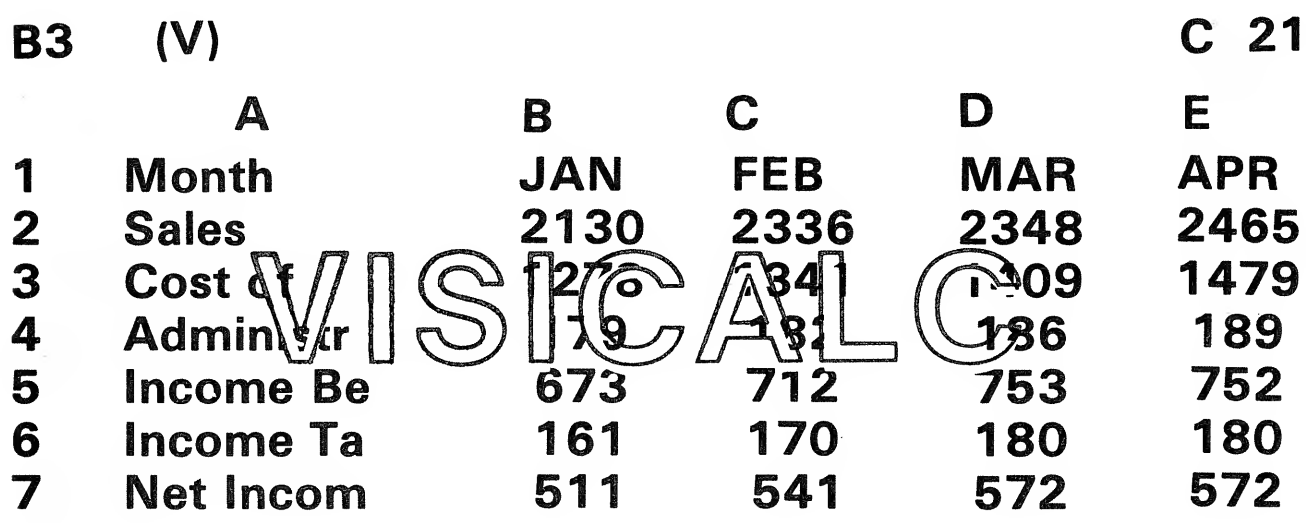
B. Master Reference Manual Only
$25



* TRS-80 is a registered trademark of Radio Shack, a Tandy Company.

LOBO drives INTERNATIONAL

LOBO DRIVES, INT'L
354 South Fairview Ave.
Goleta, CA. 93117
(805) 683-1576



(Continued from page 17)

Throw away those pencils and erasers, VisiCalc has arrived!

VisiCalc turns your TRS-80 into a "what if?" machine. What if your top salesman takes off for Topeka; what if your sales expand by over 40%; what if you raise prices by 10%; what if you pay that bill semi-annually rather than monthly? Any decision made today has an impact on the future and this routine allows forecasting to be made with ease while eliminating some of the guess-work. Obviously, any planning decision requires personal judgement and accurate information, but VisiCalc will give you the analysis and projections needed to start making those decisions. Realizing in April that increasing cost factors will substantially reduce income in December allows you to take effective measures before the fact, not when it's too late.

Here is a simple problem that is right down VisiCalc's alley (from *Small Systems World,* Dec 1980, page 24). Assume you sell 3000 items at $6.00 per item. With an $8000 variable cost, what is the maximum sales loss you can afford after raising the price to $6.60 and still maintain the same gross profit? (458 items, of course). Answering this and other "what if?" questions is just part of VisiCalc's versatility. It is a phenomenal example of useful computer software. Its flexibility and utility is unlimited and can be used by anyone in a matter of a few hours.

VisiCalc for the TRS-80 is supplied on a system disk (for 32K and up) and the program is loaded from DOS READY with the command "VC". Once VisiCalc is loaded, you have in your hands the largest calculator you have ever seen.

VisiCalc gives you a huge columnar pad of 254 rows and 63 columns - that's right, over 16,000 entries. The VisiCalc routine allows you to enter, delete, compute, graph, recombine, and in fact, do almost anything to the data of the matrix. It can be used in business for income/expense projections, budget analysis, financial statements, etc. Imagine being able to project when new employees will be needed or how much floor space is needed in August 1982. Educators can use it for grade maintenance, analysis of mathematical functions, curve fitting and graphing. Personal or home uses, such as balancing a checkbook, maintaining tax records, budget forecasting, recording stock/bond transactions are all possible. That trip to Hawaii may be possible once debt repayment schedules are altered. It is a program begging to be used!

Upon initialization, the screen shows columns labeled A to F and rows labeled 1 to 12. The screen is a small view of the complete matrix, and is referred to as the "window". To view other entries, all you do is depress the arrow keys. All work is command or cursor controlled with automatic keystroke repeat. The program is written in machine language and is quite quick to respond. At the upper left of the screen is entry A1 and the bottom-most right entry is BK254. Each one of these data cells can be given a value, a label (alphanumeric title), or defined by a formula which refers to other values.

Above the window are three lines called the "control panel". The control panel tells you the current cursor location, its label or value or formula, and includes an edit line whereby you may alter data or command other operations.

Figure 1 is an example of some data that has been entered for a small consulting business. Notice how all entries in column A are labels, and each row represents a different category. The columns are for each quarter of operation and total for the year.

The VisiCalc routine not only works in the cell that is altered, but *automatically changes all other entries that involve that cell.* If a piece of data is not available (NA) then it can be so entered. Notice that since cell E4 is NA, then so is the data for cells F4, F11 and F12. Once the number 587.00 is entered into marketing costs for the fourth quarter, all data is recomputed. F4 becomes 2235.23, F11 becomes 6517.35 and F12 becomes 987.65 as fast as you can press the enter key.

Now, watch what you can do. Assume that the following increases occur during the next year: Office +4% each quarter, travel +12% each quarter, income +10% each quarter and that the other factors remain constant. Here are the results based upon those changes on the 1980 data (See Figure 2).

You could just have easily computed a sliding average or other combination of data upon which to base your predictions. The choice is yours.

VisiCalc uses a hierarchy command structure. Within each command group is another set of options. You may back out of a command sequence by depressing the clear key. The major commands and their functions are: (Note that any row command is also possible on columns).

1. Global - gives access to commands that affect the complete matrix. Examples are setting of column width, data formatting, recalculation, restructuring to compute either first by rows or by columns.

2. Replicate - copy one row to another, use the relationship between two rows to compute a new row.

3. Format - specify the way in which data is to be displayed, either as integers, dollars/cents, graphics (asterisks), left or right justify, general (as entered).

4. Window - opens up access to viewing all of the matrix, a small part, two parts at once, or more if desired. This is the fascinating feature and gives you an easy way to see parts of the data matrix when you wish to compare sections.

5. Move - moves rows to new locations and moves other data up to fill in the blank that was created. All formula references are changed to reflect the new data locations.

6. Storage - output of data to disk, input from disk, delete data file from disk. It is also possible to merge data files, save only part of the matrix or load into a specified area when data is stored and then re-entered. When data is stored and re-entered, VisiCalc picks up right where you left off, cursor in the correct spot, all formula remembered, etc. VisiCalc has built-in ability to recall file names from the disk directory so you can check file names or disk being used before executing a store command. Output to a printer and passwords are also allowed.

Figure 3 is a graph of the function $B=A^2-2A+2$. Many of the commands listed above were used to generate what is shown. Starting with a clear matrix (C), the value -5 was entered into cell A1. A2 was assigned the value 0.2+A1, A3 was assigned 1+A2, and the replicate command (R) was used to generate all values A4 to A12. The B column was defined by a formula, (See the top left of Figure 3). Notice that parentheses must be used, VisiCalc executes from left to right unless terms are grouped. Again the replicate command (R), using a relationship based on column A and column B resulted in all entries from B2 to B12. Next, a window command (W) was used to split the screen vertically. The column width of C was set to 40 and it was put into graphics mode (F*). Finally, the replicate command was used to copy the values from column B to column C. By altering the values in column A, VisiCalc will automatically recalculate B and regraph B and regraph the results in column C.

Data manipulation is not all VisiCalc can do. There are many built-in functions for the businessman as well as the engineer. All functions and formula calculations are error trapped and any cell that contains an error in its calculation is so noted as "ERROR". It's nice not to have it lock up or quit just due to a typographical mistake. Functions are preceeded by the @ symbol to clarify that they are not to be labels. Some of the key functions are:

1. @MIN, @MAX - to find the minimum or maximum in a list or range of entries.

2. @SUM - adds up all values in a list or range of entries.

3. @COUNT - determines the number of non-blank entries in a list or range.

4. @AVERAGE is @SUM divided by @COUNT

5. @LOOKUP - scans a list for a specified entry (or greatest one lower than) and returns the matching value from a second list. This is useful in data tables.

ABS, INT, LN, LOG, EXP, SIN functions are also available. The value of PI is also pre-defined by the program.

The package comes in a 3-ring leatherette binder with a 173 page reference/instruction book. The manual has a four lesson introduction which takes you through VisiCalc's paces. Over 70 pages are devoted to command references and a discussion on parameters, formatting, structure and syntax. A complete index as well as a reference card are included. The lessons are well written and are easy to follow; nothing happens without you being told what, why and how.

This program behaves as advertised. VisiCalc has been available for some time for the APPLE computers, and there is even a user group (VisiCalc Users Group, PO Box 1523, White Plains, NY 10602). What a recommendation for a piece of software! Who knows, maybe APPLE and TRS-80 users will learn to talk to one another. ●

(VisiCalc is available from Radio Shack for the Model I and Model III at $99.95.)

C11 C 21

	A	B	C	D	E	F
1	PERIOD	1ST QTR	2ND QTR	3RD QTR	4TH QTR	TOTAL 80
2						
3	OFFICE	540	540	540	540	2160
4	MARKETING	623	457.34	567.89	NA	NA
5	PRINTING	75.56	78	109	67.56	330.12
6	CLERICAL	150	150	150	150	600
7	TRAVEL	56	78	357	92	583
8	BOOKS/JRN	150	147	125	187	609
9						
10	INCOME	1800	1765	2340	1600	7505
11					TOTAL EXP	NA
12					PROFIT	NA

Figure 1

I11 C 21

	A	H	I	J	K	L
1	PERIOD	1ST QTR	2ND QTR	3RD QTR	4TH QTR	TOTAL 81
2						
3	OFFICE	561.6	584.064	607.4266	631.72	2384.814
4	MARKETING	623	457.34	567.89	587	2235.23
5	PRINTING	75.56	78	109	67.56	67.56
6	CLERICAL	150	150	150	150	150
7	TRAVEL	59.92	83.46	381.99	98.44	623.81
8	BOOKS/JRN	150	147	125	187	187
9						
10	INCOME	1980	1941.5	2574	1760	8255.5
11					TOTAL EXP	5648.414
12					PROFIT	2607.086

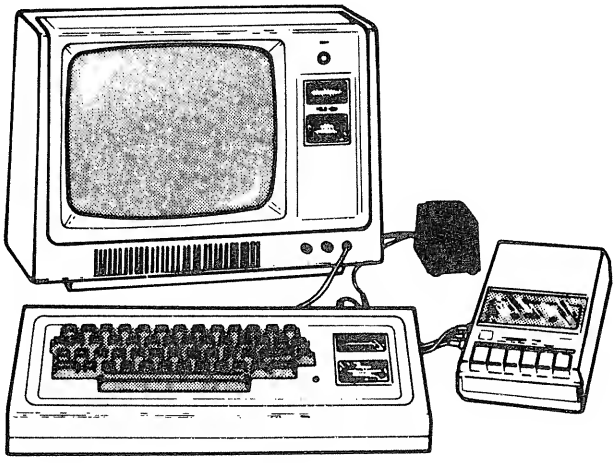
Figure 2

B1 (V) +A1*A1-(2*A1)+2 C 21

	A	B	C
1	-5	37	1.**
2	-4.8	34.64	2.**********************************
3	-3.8	24.04	3.************************
4	-2.8	15.44	4.***************
5	-1.8	8.84	5.********
6	-.8	4.24	6.****
7	.2	1.64	7.*
8	1.2	1.04	8.*
9	2.2	2.44	9.**
10	3.2	5.84	10.*****
11	4.2	11.24	11.***********
12	5.2	18.64	12.******************

Figure 3

FINE TOOLS FOR YOUR TRS-80*



UTILITY I

Here's software utility that can make programming easier for you. These powerful tools will take much of the frustration out of essential operations. You'll wonder how you managed without them. They'll stamp your work with the mark of professionalism.

RENUM—Now you can easily renumber any Level II program to make room for modification or to clean up the listing. (Only for use with 16K of memory.)

DUPLIK—This program will let you duplicate any BASIC, assembler, or machine-language program, verify the data, and even copy Level I programs on a Level II machine.

See how efficient and well-ordered your programming becomes. (T1)

Order No. 0081R $9.95.

UTILITY II

One of the most popular and useful of the utility packages, Utility II is the second in a series of programs designed to take the drudgery out of editing.

This combination of programs adds flexiblity to your BASIC programs by allowing you to combine them with other BASIC programs, or with machine-language programs or routines.

CFETCH—You'll be able to merge BASIC programs, with consecutive line numbers, into one program. CFETCH can also search through any Level II program tape and display the file name for all the programs.

CWRITE—Combine subroutines that work in different memory locations into one program. CWRITE works with BASIC and/or one or more machine-language programs. It will even give you a general checksum to verify that your program hasn't dropped any bits.

Use these two programs as your dependable "assistants" to speed merging processes and as a totally reliable retrieval means in search and edit operations.

A fine tool for your TRS-80! (T1)

Order No. 0076R $9.95.

COMPRESSION UTILITY PACK

Do you want to add sound routines to a Space Trek program that already uses 16K? Or maybe you need an extra column in that financial report program, but when you run it, you get OM errors?

With a wave of your hand, and a little help from either of the COMPRESSION programs in this package, your problems are over.

COMPRESS-80—fits in 265 bytes, deletes spaces, and offers the choice of leaving REMark line numbers in the program, or deleting them altogether.

SUPERCOMPRESS—uses 767 bytes and can do everything COMPRESS-80 can do PLUS, it packs the program into the smallest possible number of multiple statement lines.

With the Compression Utility Pack and your own programming skill, you can add all those little extras to your BASIC programs. (T1)

Order No. 0246R $19.95.

ULTRA-MON

ULTRA-MON is a unique and powerful machine-language monitor. It is ROM independent and will function in Level II or DOS BASIC. With ULTRA-MON, you will be able to write, modify, study and debug machine-language programs. Plus, you'll be able to avoid the frustrations and "bomb-outs" usually associated with machine-language programming. ULTRA-MON displays, disassembles, traces (hardcopy trace disassembly, too!), modifies, relocates memory, prints and even relocates itself with simple commands. Using interpretive execution, ULTRA-MON allows you to put breakpoints in ROM. This powerful monitor can even fetch, decode, disassemble and analyze each instruction individually so that your program cannot bomb out.

ULTRA-MON is designed for the beginning machine-language programmer as well as the professional. The documentation contains a Simple Demonstration section geared to the novice programmer. Consequently, the program is a learning device as well as an extremely useful programming tool.

If you are serious about programming, you need to add this powerful utility to your library today. (T1)

Order No. 5003R $24.95.

TLDIS & DLDIS

You've bought a super machine-code program, but now wonder how it works. Maybe you even used a quick PEEK routine to glance through it when it was in memory. If so, you definitely noticed the complete lack of comments in the code, making it almost impossible for you to decipher and understand it.

Well, Instant Software's Labeling Disassemblers are the answer to your problem.

TLDIS (Tape-based Labeling DISassembler) and DLDIS (Disk-based Labeling DISassembler) are three-pass, label-assigning disassemblers which assign labels (where appropriate) to the routines in a machine-language program. Their output is almost identical to that of a hand-assembled source code.

You can send the disassembly to a line-printer (Radio Shack parallel port) for either TLDIS or DLDIS. (The difference between these utilities is the storage mode of the disassembly.)

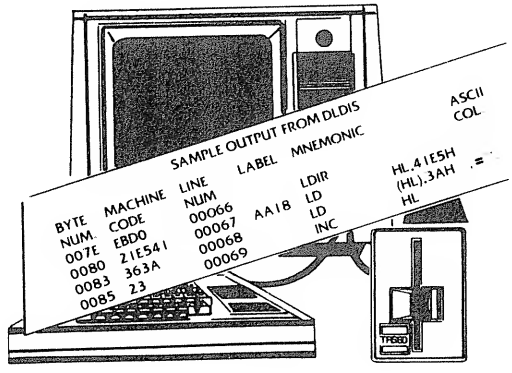
TLDIS can send the disassembly to cassette tape, DLDIS can send it to disk; both send it to the video monitor. The stored disassembly from TLDIS may be reassembled with Radio Shack's EDTASM™—the disassembly from DLDIS, with Apparat's extension of EDTASM™. Because of the use of labels, it is a simple matter to change any object code program by disassembling it and then making changes to the resultant source code, without losing track of jump/load addresses. Labels start with "AA00" and increment up, in even numbered steps (AA02, AA04, etc.). The odd numbers (AA01, AA03, etc.) are left for you to use for the source code during reassembly.

The printing of the disassembly may be temporarily halted by using [SHIFT] @ (just as in BASIC) or it may be ended by pressing the [BREAK] key. It also has a comments column to display ASCII characters used in a LD or CP opcode.

TLDIS and DLDIS may be relocated in memory to avoid conflict with the program you disassembler.

The next time you need to "climb inside" a machine-code program, take DLDIS or TLDIS with you. We promise that it will be an easier journey.

Order No. 0230R (TLDIS) $14.95 (T1)
Order No. 0231RD (DLDIS) $19.95 (T2)



* A trademark of Tandy Corporation

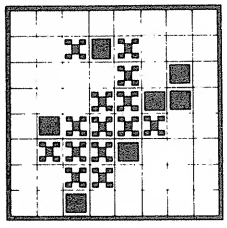
Instant Software™

PETERBOROUGH, N.H. 03458
603-924-7296

Proven Favorites. . . These 10 Popular Games Offer Exciting New Challenges

Here are 10 games that have never lost popularity. They've been played in many forms for many years. And now they're among the favorite software packages in the computerized home-entertainment field.

Don't miss out on the excitement, challenge and downright fun of these programs. Take the word of thousands of satisfied players. They're still tops. Order yours today and play your kind of game.

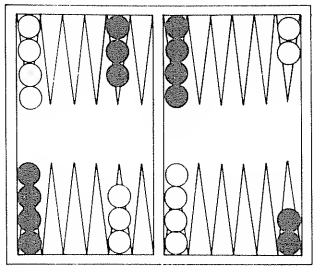


TRS-80*

OTHELLO

OTHELLO—Pit your strategic powers against a merciless, computerized opponent. You play on a board of 64 squares. When you capture your opponent's game disks (by bracketing them with your own disks), they immediately change sides to become members of your set. You can capture several game pieces at one time as long as they are in-line horizontally, vertically or diagonally. Major reversals of score are commonplace as whole blocks of men change side in an instant. **(T1) Order No. 0046R $9.95**

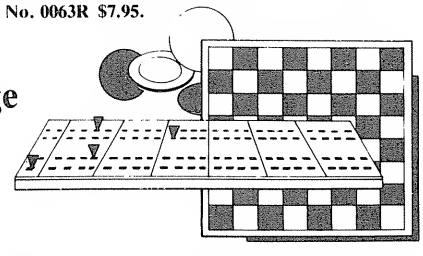
Beginner's Backgammon and Keno



BEGINNER'S BACKGAMMON/KENO—Why sit alone when you can play these fascinating games: ● BACKGAMMON: Play against the computer in a game that's sure to sharpen your skills; ● KENO: Enjoy this popular Las Vegas gambling game—guess the right numbers and win big! **(T1) Order No. 0004R $7.95.**

CARDS—A one-player package to let you play, with your computer, these famous games: ● DRAW AND STUD POKER: These programs will keep your game sharp; ● NO-TRUMP BRIDGE: Develop your strategy and (hopefully) increase your skill. **(T1) Order No. 0063R $7.95.**

Your Cribbage and Checkers Partner



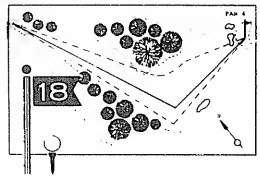
YOUR CRIBBAGE AND CHECKERS PARTNER—● CRIBBAGE is a two-person game that you are sure to enjoy. This is NOT a tutorial—it is a game worthy adversary. ● CHECKERS: An old favorite which follows international rules, including multiple jumps. **(T1) Order No. 0068R $9.95.**

CHESSMATE-80—This versatile chess opponent gives you a choice of ten levels of play, from the "blitz" level (the computer has 3 seconds to move) to the infinity level (where the computer will consider every possible move—which could take years). This machine-language program is a conservative player and follows all the rules of international play. CHESSMATE-80 can teach you how to move and allow you to set up the board and play end games or special problems. CHESSMATE-80 battled Sargon II to a draw at two minutes a move and beat Microchess 1.5 in six moves. **(T1) Order No. 0057R $19.95.**

BOWLING—Let your TRS-80 set up the pins and keep score. One player can pick up spares and get strikes. **(T1) Order No. 0033R $7.95.**

Apple**

GOLF



GOLF—Without leaving the comfort of your living room you can enjoy a challenging 18 holes of golf with a complete choice of clubs and shooting angles. Now, you need never cancel your game because of rain. One or two players can enjoy this game on your Apple II with Applesoft and 32K. **Order No. 0018A $7.95**

PET***

CASINO I—Gamblers, gather round for: ● BLACKJACK—a fun-to-play and tutorial version that allows you to play every combination that they play at the MGM Grand Hotel in Las Vegas; ● ROULETTE—also a game *and* a tutorial which could enable you to develop the skill to make a fortune at the tables. **Order No. 0014P $7.95**

CASINO II—This is not simply a craps game, it is a tutorial program that will teach you the odds on every type of bet, so you can steer clear of the sucker bets that impoverish the unwary. Using the exact rules used at the MGM Grand Hotel in Las Vegas, you'll play under actual gambling house conditions. Get the edge you need to "get lucky." **Order No. 0015P $7.95**

CHECKERS & BACCARAT—Using International Rules, you'll play ● CHECKERS with your computer and the computer will keep score, time the moves and even tell you when to jump (it'll be so busy, you may have an edge!); ● BACCARAT—gives you the choice of playing Las Vegas or Blackjack style and the computer will deal the cards, figure the payoff and keep track of your bankroll. **Order No. 0022P $7.95**

ORDER YOUR INSTANT SOFTWARE TODAY!

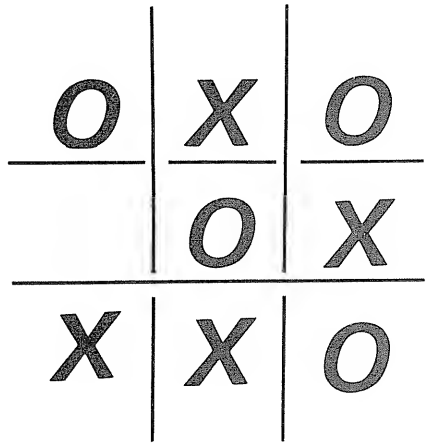
Just Call Toll-Free 1-800-258-5473

(T1)=TRS-80 Model 1, Level II, 16K RAM

Instant Software™

PETERBOROUGH, N.H. 03458 603-924-7296

* A trademark of Tandy Corporation
** A trademark of Apple Computer Inc.
*** A trademark of Commodore Business Machines Inc.



Smart Tac Toe

A look at Artificial Intelligence

This program works without modification on Models I and III, 16K and up. Although written for disk, modification for tape is possible.

Steve Kelley
Oakland, CA

After my curiousity was aroused by several articles about "smart" games - games that can actually learn from their mistakes, I began to wonder how far this idea could be taken. In this article I am exploring the idea of a program that could not only learn from mistakes, but actually teach itself to play the game in the first place.

Artificially intelligent games have been around for some time now, and I imagine that many people have experimented with them. The programs I have seen in the past played very simple games, such as the one played with six Chess pawns, and would eventually become unbeatable. On looking at these game programs closely however, I realized that while these programs could learn from their mistakes, they lacked the ability to learn to play the game in the first place. For the program to be able to play, every possible combination of board position had to be loaded into an array with every possible subsequent move loaded into similar arrays. To pick a move, the computer would find its present position in the first array, then choose its move from the "possible moves" array. If this move results in the computer losing on the next move, the array element containing the losing move is zeroed, effectively deleting the move from the list of moves the computer can choose from. This is how the program is able to learn from its mistakes. Since each losing move is deleted from its list, the program will never make the same mistake twice. This may be the easiest way to implement a program of this nature, but it results in the programmer, rather than the computer, being faced with the task of calculating every possible board position and every possible corresponding subsequent move. Besides this being a lot of work, both the brain-work and the keying in of ump-teen DATA statements, it also "cheats" for the computer, in that the program has complete familiarity with the game before it ever plays it.

My idea was a simple one: Have the computer start with a blank memory and build a table of moves based on its actual experience. Couple this with the ability to learn from its mistakes and you have a program that is not only easier to implement, but much more flexible as well. To illustrate this and test my idea, I needed a game with thousands of game board combinations (compared to less than 50 for HEX-PAWN) that was still simple enough to be learned by the computer. Tic-Tac-Toe seemed a reasonable enough choice based on this criteria. It is simple, has many combinations, and has the added advantage that it is almost universally known.

The program to play (and learn) the game of Tic-Tac-Toe is quite simple and relatively short (see program listing). In fact, the portion of the program involved with the actual decision making/remembering is one of the smallest. Most of the program merely handles the mechanics of displaying the game on the screen, etc. The only DATA statements contain the "X" and the "O" graphics character information and data for a fast matrix manipulation routine I added. The latter is used to check to see if the game being played is a mirror image (in any direction) of some previous game. The only other decision-making routines are to determine if the human's move is valid and to check for a win after each move. Other routines are the timed input routine which starts the computer playing against the random number generator if you don't make a move for 30 seconds, and routines to load and save the cumulative memory to disk (these can be modified to save and load to tape instead).

The program will prompt you whether to load in data from previous games once at the beginning of the program. To save current level of play simply enter "9999" as your move. The data will be saved and the program will prompt your move again.

The computer's strategy is the simplest: A) take the first empty square not zeroed in

```
'
'       *SMARTTAC*
'       *BY  STEVE KELLEY*
'
  CLS:PRINT TAB(18);"SMART-TAC-TOE"
  CLEAR 999
  DEFINT P-Z:DEFSTR A-H
  DIM P(9),Q1(9),Q2(8),Q3(2),Q4(8,9),Q5(9),A(2),P1(2,1023)
  FORZ=0TO2:FORZ1=1TO10:READZ2:A(Z)=A(Z)+CHR$(Z2):NEXTZ1,Z
  FOR Z=1 TO 9:READ Q1(Z):NEXT Z:E1=CHR$(30)
  FOR Z=1 TO 8:READ Q2(Z):NEXT Z
  FOR R2=1 TO 8:FOR X=1 TO 9:READ Q4(R2,X):NEXT X,R2
  FOR X=1 TO 9:READ Q5(X):NEXT X:Q0=-1
  PRINT:PRINT"USE DATA ON DISK ?   1=YES  2=NO ";:GOSUB 880
  IF VAL(A)<>1 THEN Q9=0:GOTO 160
  OPEN"I",1,"DATATAC/TXT":INPUT#1,Q0,Q1,Q2,Q3,Q4,Q5,U1
```

```
150 FOR R=0 TO 1023:INPUT#1,P1(0,R),P1(1,R),P1(2,R):NEXT R:CLOS
    E
160 PRINT:PRINT"ENTER "+CHR$(34)+"9999"+CHR$(34)+" AS MOVE TO R
    ECORD NEW DATA TO DISK."
170 INPUT"HIT (ENTER)  TO CONTINUE .  .";C
180 '             *NEW GAME*
190 QG=QG+1:Q3(1)=0:Q3(2)=0:FOR X=1 TO 9:P(X)=0:NEXT X:PB(1)=0:
    PB(2)=0
200 '           *MAIN LOOP*
210 IF POINT(21,26)<>0 THEN 250
220 CLS:FOR Z=1 TO 60:SET(Z+20,14):SET(81-Z,26)' *DRAW CROSSHAT
    CH*
230 SET(60,Z/1.9+5):SET(40,37-Z/1.9):NEXT Z
240 PRINT@183,"1  2  3";:PRINT@247,"4  5  6";:PRINT@311,"7  8
    9";
250 FOR X=1 TO 9:PRINT@Q1(X),A(P(X));:NEXT X' *PRINT PIECES*
260 PRINT @0,"GAMES PLAYED";QG;" I WON";QC;" YOU WON";QH;" LEVE
    L OF PLAY";U1;
270 FOR Z1=1 TO 2:FOR Z=1 TO 8' *CHECK FOR WIN*
280 IF (PB(Z1) AND Q2(Z))=Q2(Z) THEN Q3(Z1)=Z
290 NEXT Z,Z1:ON SGN(Q3(1))+SGN(Q3(2))*2 GOTO 650,730
300 IF PB(1)+PB(2)=511 THEN 790' *CHECK FOR DRAW*
310 IF QT=0 THEN QT=1 ELSE QT=0:GOTO 430
320 '           *HUMAN'S TURN*
330 C=INKEY$:IF C<>"" THEN QS=0' *RETURN CONTROL TO KEYBOARD ?*
340 PRINT@896,E1+"MAKE YOUR MOVE  ";
350 S=RND(9):IF QS=0 THEN GOSUB 880:S=VAL(A)
360 IF S<>9999 THEN 390' *CODE FOR RECORD ON DISK ? *
370 OPEN"O",1,"DATATAC/TXT":PRINT#1,QG;QH;QC;U1
380 FOR R=0 TO 1023:PRINT#1,P1(0,R);P1(1,R);P1(2,R):NEXT R:CLOS
    E
390 IF S<1 OR S>9 THEN 330
400 IF P(S)=0 THEN P(S)=1:PB(1)=PB(1) OR Q5(S):GOTO 210
410 PRINT @896,E1+"NO NO NO . . .";:FOR Z=1 TO 500*(1-QS):NEXT
    Z:GOTO 330
420 '           *COMPUTER'S TURN*
430 PRINT @896,E1+"MY TURN ";
440 R2=8
450   PB(1)=0:PB(2)=0:FOR X=1 TO 9:PB(P(Q4(R2,X)))=PB(P(Q4(R2,X
    ))) OR Q5(X):NEXT X
460   Q=PB(1)+PB(2)+1:R1=2*Q+1' *GENERATE HASH CODE*
470     IF PB(1)=P1(1,Q) AND PB(2)=P1(2,Q) THEN 500
480     IF P1(1,Q)=0 AND Q<>1 THEN R2=R2-1:IF R2>=1 THEN 450 EL
    SE 580
490    Q=(Q+R1) AND 1023:GOTO 470
500   IF P1(0,Q)=511 THEN 650' *CONCEDE IF NO MOVES*
510 '
520   FOR Z=1 TO 9:IF (P1(0,Q) AND Q5(Z))=0 THEN X=Z:Z=9
530   NEXT Z
540   P(Q4(R2,X))=2:Q3=Q:Q2=X' *PUT O IN CURRENT POSITION*
550 PB(1)=0:PB(2)=0:FOR X=1 TO 9:PB(P(X))=PB(P(X)) OR Q5(X):NEX
    T X
560 GOTO 210
570 '           *SIMILAR MOVE NOT FOUND*
580 U1=U1+1:P1(0,Q)=PB(1) OR PB(2):P1(1,Q)=PB(1):P1(2,Q)=PB(2)
590 '
600 FOR Z=1 TO 9:IF (P1(0,Q) AND Q5(Z))=0 THEN X=Z:Z=9
610 NEXT Z
620 P(X)=2:PB(2)=PB(2) OR Q5(X):Q3=Q:Q2=X
630 GOTO 210
640 '           *COMPUTER LOSE*
650 P1(0,Q3)=P1(0,Q3) OR Q5(Q2)' *MASK OFF LOSING MOVE*
660 QH=QH+1
670 GOSUB 830
680 PRINT @896,E1+"I LOSE ";:IF QH<>1 THEN PRINT" AGAIN ";
690 IF U1>10 PRINT"- BUT I'M LEARNING ";
700 IF QS=0 THEN GOSUB 880
710 GOTO 190
720 '           *HUMAN LOSE*
730 P1(0,Q3)=(NOT(Q5(Q2))) AND 511' *MASK OFF NON-WINNING MOVES
    *
740 QC=QC+1
750 GOSUB 830
760 PRINT@896,E1+"I WIN ";:IF QC<>1 THEN PRINT"AGAIN ";
770 GOTO 700
780 '           *KAT'S GAME*
790 IF QC>QH THEN C="YOU" ELSE C="I"
800 PRINT @896,E1+"AT LEAST "+C+" DIDN'T LOSE !! ";
810 GOTO 700
820 '           *DISPLAY WIN*
830 FOR Z=1 TO 7*(1-QS)
840   FOR X=1 TO 9:IF (Q5(X) AND Q2(Q3(1)+Q3(2)))<>0 THEN PRINT
    @Q1(X),A(0);
850   NEXT X:FOR X=1 TO 9:PRINT@Q1(X),A(P(X));:NEXT X
860 NEXT Z:RETURN
870 '           *TIMED INPUT ROUTINE*
880 A="":PRINTCHR$(14);
890 FOR Z=1 TO 3000:B=INKEY$:IF B<>"" THEN Z=10000
900 NEXT Z:IF B="" THEN QS=1:GOTO 950
910 IF B=CHR$(8) AND LEN(A)>0 THEN A=LEFT$(A,LEN(A)-1):PRINT B;
    :GOTO 880
920 IF B=CHR$(13) THEN PRINT:GOTO 950
930 IF ASC(B)>=48 AND ASC(B)<=57 THEN PRINT B;:A=A+B
940 GOTO 890
950 PRINTCHR$(15);:RETURN
960 '           *DATA FOR CHARACTORS*
970 DATA 32,32,32,26,24,24,24,32,32,32
980 DATA 137,152,129,26,24,24,24,134,130,132
990 DATA 151,131,149,26,24,24,24,141,140,133
1000 DATA 142,152,162,398,408,418,654,664,674
1010 DATA 7,56,448,73,146,292,273,84
1020 '           *DATA FOR MATRIX INVERSIONS*
1030 DATA 1,2,3,4,5,6,7,8,9,3,2,1,6,5,4,9,8,7
1040 DATA 1,4,7,2,5,8,3,6,9,3,6,9,2,5,8,1,4,7
1050 DATA 9,8,7,6,5,4,3,2,1,7,8,9,4,5,6,1,2,3
1060 DATA 7,4,1,8,5,2,9,6,3,9,6,3,8,5,2,7,4,1
1070 '           *DATA FOR POWERS OF TWO*
1080 DATA 1,2,4,8,16,32,64,128,256
```

array. B) concede if they are all zero. C) never make the same mistake twice.

You will find from just a few games that the computer is very quick to learn defenses to your particular strategy, even though its trial and error method makes it look pretty stupid. The computer is not programmed with any knowledge of the game Tic-Tac-Toe or with any strategy to win or keep from losing, only how to detect a win. This can be demonstrated by changing the win detection routine to check for three in a corner being the only way to win. The computer will learn to beat you just as fast this way or any other way as it did in the original game.

Food for Thought

The whole idea of artificial intelligence involves some fascinating concepts. One is that the program has the ability to learn to play better than its programmer. A not-so-simple case of the pupil passing the teacher. Another is that since the computer is learning to play from your play, it will compliment your particular strategies. If you continuously use your sure-fire best way to win, it will soon block you every time. Taking this a bit farther, after you can no longer beat the computer, let someone else who is pretty good at the game try. How about that, they did it! This is because the computer is only ready for your method of winning. Very soon though, the newcomer can no longer win, nor can anyone else who plays well. Now let a youngster play the computer, and watch them win! The computer is ready for every triple-corner two-way sure-win strategy, but not for someone who simply places three X's in a row.

Games Played vs Learning

While playing against the program, I learned a few things from the computer about Tic-Tac-Toe. For instance, when I first learned the "secret" of being able to beat anyone who didn't know the secret, I was told to start in a corner whenever I started first. The worst I've ever done this way was to tie. I didn't think it was possible to lose using this method knowing the counters I know, even against an expert. However, after playing by itself for quite a while, the computer always started in the center square. This can only mean that there is a way to be beaten every time for every other starting move! This was discovered after about 50,000 games played 24 hours a day while the computer was not in use.

The statistics for the learning level (number of different games played) and the ability to be beaten are quite remarkable. When I could no longer beat the machine without considerable effort, the learning level was up to about 48. My wife Barbara, who plays better than I, pushed this figure up to about 60. I assumed from these amounts that the maximum learning level must be about 80 or 90. Left playing by itself the first night, the computer got the learning level up to 512, which completely filled the HASH table and stopped the program! This was totally unexpected, since we had played manually for about 4 hours and only got to 60, and especially since the computer can look at each game from so many angles to check for a similar game in memory. Final figures are 25,000 games played, learning level well over 600, games lost by computer 2500. These figures are for playing against the random generator, and would be much different if played against a human opponent (or another artificially intelligent computer?).

Practical Applications

OK, now it plays an unbeatable game of Tic-Tac-Toe, tomorrow the world? Not quite. I've tried everything from a self-adjusting inventory program to a routine to give advice on betting at draw poker, but the program just learns too slow. It took 25,000 Tic-Tac-Toe games to get good at that, so just picture an inventory controller that made 25,000 mistakes, or a poker advisor that gave you 25,000 bum steers. Got the picture? So here is the basic tool for computer learning. It's not perfect, or even very practical right now, but I know there will come a time when we will see a program that will figure your income tax while automatically and dynamically re-adjusting itself to your exact needs, based on some kind of artificial intelligence. ●

(Note: Smart Tac-Toe is available on cassette with the cumulative memory of 25000 games and complete documentation for $6.00 ppd. Steve Kelley, 9506 Peach St., Oakland, CA 94603)

Anatomy of the Program SMART TAC TOE

R C BAHN

1. INTRODUCTION

This is an artificial intelligence program which "learns" to play tic-tac-toe as described in the accompanying text. The program assumes a rather bureaucratic concept of success by primarily avoiding failure.

Eventually, against a random opponent, the computer will win about 75% of the games and tie the remainder. This amounts to winning essentially all games in which the computer has the first move and winning half the games in which the computer has the second move. When the computer is adequately trained, usually all games will be tied against an alert knowledgeable human opponent.

This program provides two types of instructive activities. The first is the study of the automated "learning" process. The second is the study of the programming techniques.

To get to the study of the automated "learning" process, copy the listing of the program very carefully. Every parenthesis, semicolon, comma, apostrophe, line number and line arrangement is important. Once the program is running, skip to the section of this article entitled "Evaluation of the Learning Process."

The program listing is relatively short but very compact. The flow of the program is circuitous. However, there are numerous comments which help to divide the program into logical modules.

To fully understand the programming techniques one should be familiar with the following topics: DATA statements, concatenation of string variables, subscripted variables, FOR-NEXT loops, logical IF statements, subroutines, RND statements, binary numbers, binary to decimal conversion and particularly Boolean logical operations.

II MAJOR VARIABLES

Variables with beginning letters of A-H are string variables. The remaining variables (P-Z) are integers. These definitions can be found in line 50 of the program.

The most important string variable is A(N). A(N) is formed in line 70. A(0) is a blank. A(1) is the screen symbol for "X", A(2) is the screen symbol for "0". These symbols are formed from the data associated with lines 960 to 990 in the DATA statements. In line 70:3 the individual numbers in the data are read as Z2. In line 70:4 each of these are concatenated as an ASCII character to form the respective A(N). To check this part of the program, copy lines 10-70 and lines 960 to 990. After you have RUN these few statements, press BREAK and type the following "one-liner" program in the direct mode using no statement number. FOR I=0 TO 3:PRINT A(I):NEXT I. The three symbols will be displayed on the video screen.

Several integer variables deserve description. QC is the number of games won by the computer. QH is the number of games won by the opponent. U1 is the number of new games studied. QG is the total number of games played. The number of ties=QG-(QH+QC). QC, QG, QH, and U1 are continuously displayed at the top of the screen.

Q1(I) is indexed 1 to 9 and consists of the screen addresses of the nine possible positions of the screen symbols. Q2(I) is indexed 1 to 8 and consists of the decimal representations of the binary coding of the eight winning positions. Q1(I) and Q2(I) appear in lines 1000 and 1010 respectively.

Q4(I,J) is an 8 x 9 array containing the eight possible positions of the tic-tac-toe figure if it were rotated 180 degrees around the vertical axis, the horizontal axis, or the major and minor diagonal axes. This array is used in tests for geometrically similar moves. The data appears in lines 1020 to 1060.

The tic-tac-toe figure is composed of nine positions. Unique coding of the occupancy of nine squares numbered 0, 1, 2, 3, 4, 5, 6, 7, 8 can be obtained by assigning each the value of its respective power of two, i.e. 1, 2, 4, 8, 16, 32, 64, 128, 256 and then adding the assigned numbers of the occupied squares. The greatest number will be 2^9-1 or 511. The powers of 2 are stored in Q5(I) and appear in line 1080 of the data. It is more rapid and precise to look up the powers of two than to compute them each time they are needed.

Within the program the decimal coded binary information is used with the array B1(I,J). This matrix is dimensioned in line 60 of the program (B1(2,1023)). In the first dimension, 0, 1, and 2 are all used. The index of the second dimension is formed as Q in line 460 and consists of one plus the sum of coded occupancies by the first and second move players. The latter data is stored in PB(1) and PB(2) respectively and updated during the play of the game.

The body of data within B1(I,J) is first generated for a specific value of Q in line 580. Note that P1(1,Q)=PB(1) and P1(2,Q)=PB(2). Finally P1(0,Q)=PB(1) OR PB(2). This is a

logical OR. The data within P1(I,J) is later modified by OR and AND logic in lines 650 and 730 upon the events of losing or non-winning moves, respectively.

Throughout the program, arrays P1(I,J) and Q5(I) are queried by IF statements using Boolean operators to determine legal moves, prior similar moves, next allowable moves, and winning positions. These maneuvers occur in lines 280, 400, 450, 490, 510, 520, 550, 580, 590, 600, 620, 650, and 840.

The use of Boolean operators is elegant and rapid. You will have mastered a significant area of computer science by understanding them. Consider line 280. A specific player's combination of moves is stored in PB(Z1). A winning combination is stored in Q2(Z). The question is asked: Is the winning combination of Q2(Z) a member of the set of positions in PB(Z1)? If the answer is yes, the logical AND will return the number stored in Q2(Z), otherwise some other number or zero will be returned. In essence, the logical AND returns members, or in generalized terms, the binary bits, which are in common.

The logical OR is illustrated in line 580. A similar move was not found. U1 was incremented. Next, P1(0,Q)=PB(1) OR PB(2). P1(0,Q) now contains a record of the occupancy of the board irrespective of first or second move. In essence, the logical OR returns members, or in generalized terms the binary bits, which occcur in the first or the second or both tested configurations. Since occupancy of the tic-tac-toe board is mutually exclusive between players, simple addition would result in the same outcome in this situation, but not necessarily in the other uses of the logical OR within this program.

III EVALUATION OF THE AUTOMATED "LEARNING" PROCESS

For any automated "learning" process, one would like to know the expected limits of performance and the time necessary to actually achieve a performance which is close to the expected limits. The beginning of such an evaluation consists of collecting the raw data which is displayed on the first line of the video screen. One could surmise that very little additional "learning" would occur if the level of play and the number of games won by the human did not change for many iterations.

In addition to these criteria, one can construct a standardized "learning curve" which will allow you to more easily compare various experiments you might conduct. The total number of trial (QG), the number of wins by the computer (QC), and the number of wins by the human opponent (QH) could continue to increase.

The percentage of games won by the computer (QC/QG)*100, the percentage of games won by the human (QH/QG)*100 and the percentage of ties (1-(QC+QH)/QG))*100 will behave in a more controlled fashion. The percentages will sum to 100%. The percentage of games won by the human will approach zero. The percentage of ties appears to approach about 25%. The percentage of games won by the computer appears to approach about 75%. These limiting long term values or asymptotes are only approximations since they were not rigorously obtained by theoretical analysis of the problem but rather determined by observation of two experiments, one running for 12,401 iterations and the other running for 67,626 iterations.

You can study your own computer's learning curve by tabulating the raw data at appropriate intervals, computing the percentages for each interval and plotting the percentages on the Y axis against the total number of trials on the X axis. This might even be a separate programming problem.

Expected performance values can be computed as a function of QT, the total number of trials. An empirical expected average learning curve for percentages of ties can be approximated by the function Y_T=100*(.25*(1-EXP(A*QT))). For percentages of computer wins Y_C=100*(.5+.5*(1-EXP(B*QT))). For percentages of human wins Y_H=100-Y_T-Y_C. In the two preliminary experiments, the coefficient A was about -.005(-.002 to -.008). The coefficient B as about -.007(-.006 to -.008). A an B reflect in one number the rate of "learning".

With these procedures you are now ready to evaluate the "learning" process. The greatest action occurs during the first 2000 iterations. Against a purely random opponent the experimental curve will not exactly follow the average expected curve but will develop plateaus. Different beginnings may be obtained by randomizing the seed of the random number generator during program initialization.

What is the effect of early or late training by a human opponent? What is the last "trick move" to be defensed by the computer after prolonged random learning? How many random trials are enough? How many human trials against an expert opponent are enough?

Divide your experiments into two stages and call the information in the first stage the training set and the information in the second stage the test set. During the training set, tabulate the raw data at appropriate intervals in the usual manner. Continue the process during the test set but record the raw data at the end of the training set and subtract these numbers from the raw data displayed at the end of various intervals during the test set. Compute the preformance percentages of the test set on the basis of these differences. Can human play as a training set provide all the information necessary for maximum winnings against a random opponent in a test set? How can the program be improved to make the percentage of computer wins against a random opponent approach greater than 90% and still maintain the philosophy of the program that the list of specific winning combinations is not directly accessible to the decision-making procedure?

If you do multiple experiments you may want to save each file. This can be done by renaming the disk files in the system mode with the RENAME command. However, remember to make the name of the disk file to be used in the program identical to the name of the file to be called by the program. In the program listing this name is "DATATAC/TXT".

This program is written very efficiently and runs at about 240-300 iterations per hour. Nevertheless, the generation of over 1000 iterations becomes very tedious. The program can be compiled in machine-language with a BASIC compiler. The prolonged experiments described earlier were performed by a compiled program at a rate of 1500-2000 iterations per hour. The Boolean logic within the program is ideally suited for this manuever. The result was a six to eight fold increase in the speed of execution of the program.

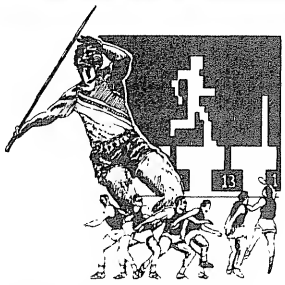
In summary, these analytic techniques begin to provide methods for answering the following questions. How does one describe the behavior of an automated learning process? How can one study the effects of varied procedural interventions? How can one determine the "best" algorithm to accomplish the desired objective of the learning process? ●

IV PROGRAM COMMENTARY

10-110	Initialization of program, see discussion of variables.
120-150	Menu and loader for disk data file.
120	GOSUB 880 is the timed input routine.
140	Open a serial file called "DATATAC/TEXT" and read variables QG, QH, QC, U1. This is the raw data displayed on the first line of the video screen.
150	Continue reading P1(I,J) which has three entries (0,1,2) for length of 1023 members; then close file.
160-170	Prompt concerning updating or writing data to disk.
180	Initialization of a new game. Line 190 is a return point from line 710.
210-240	If not present, draw game diagram on video screen and number scheme in upper right-hand corner. Line 210 is return point from lines 400, 560, and 630.
250-310	Print symbols, header, check for win, check for draw, go to computer (430) or human (320) turn.
320-350	Display human turn or take random move. GOSUB 880 is timed input routine. Any key may be pressed to return control to the keyboard.
360-390	File writing routine similar to lines 140, 150. These lines can be changed for tape files.
400-410	Routine to check for prior occupancy.
420-430	Computer's turn. Line 430 is a branch point from line 310.
430-480	Updating loop. R2 is initialized in line 440 and decremented in line 480:1. For discussion of Q, R1, PB(1), PB(2) see section II.
500-530	Note that line 510 is a REM preceded by an apostrophe. No possible allowable moves remain thus the logical AND of line 520 returns a zero. Setting Z=9 in line 520 sets up an escape from the loop in line 530.
540-560	Put 0 in current position, initialize and form PB(I), and return to main loop (210).
580-630	Similar move not found, update U1 and P1(I,J). Note line 590 is a REM. For discussion of Boolean logic see section II.
640-710	In the event of computer loss, modify P1(0,J), display win (GOSUB 830), print messages.
720-770	In the event of human loss, modify P1(0,J), display win, print messages.
780-810	Tie game routine.
820-860	This is a rapid subroutine to display wins.
840	Set up loop; blank out winning combination (A(0)).
850	End last loop; set up loop; reprint winning combination.
870-950	Timed input routine.
900	If loop is completed QS1 and automatic mode of play is started.
910-930	Tests for valid keyboard input (1 to 9).
960-1080	Data.
970	Data for "blank" symbol.
980	Data for "X" symbol.
990	Data for "0" symbol.
1000	Data for screen addresses of positions in playing array.
1010	Data for decimal coded binary representations of winning combinations.
1030-1060	Data for the arrangement of each of the 8 possible equivalent positions of playing array.
1080	Data for powers of two. ●

THE PROGRAM STORE

OLYMPIC DECATHLON



By Timothy Smith from Microsoft
The graphics capabilities you were promised when you bought your computer are finally utilized in this marvelous series of programs. Just like the real Decathlon, you compete in 10 demanding games that encompass different forms of running, jumping and throwing.

Play alone or with as many as eight competitors, the gold medal will always go to the skillful -- never the luckiest -- because your score depends entirely on skillful manipulation of the keyboard.

One of our more expensive game collections -- and worth every penny! You MUST see this system in action. Otherwise, you simply won't believe the combination of truly outstanding graphics, fast-paced action, nail-biting intensity, and even a touch of comedy you'll experience with Olympic Decathlon!

16K Tape...$24.95 32K Disk...$24.95

COSMIC FIGHTER

By B. Hogue & J. Konyu from Big Five
Terrific sound, graphics and unique challenges mark this new space game a winner! While fighting off the alien convoys, each more skillful than the last, you must keep track of your rocket fuel or risk explosion as you maneuver toward your space station. Can you dock immediately, or is the station overrun by aliens? Find out by ordering Cosmic Fighter today.

16K Tape...$14.95
32K Disk version...$17.95

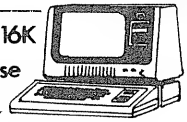
GAMMON CHALLENGER

By Ray Daly & Tom Throop from Acorn
The backgammon player featured in Personal Computing is now back in a faster, even better version! The game logic of the new Gammon Challenger has been compiled to machine language for extra speed, and there are more special features than ever.

Choose one of three levels of play, but don't get too ambitious -- Gammon Challenger will put your skill to the test at all levels. For serious players, the "doubling cube" option can be used for added excitement. There are other computer backgammon games, but none quite like Gammon Challenger.

Protected Tape...$14.95

TRS-80 Level II 16K unless otherwise noted

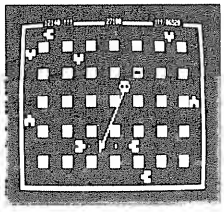


PROJECT OMEGA

By Bob Nicholas -- Adventure International
In probably the most accurate simulation ever produced for a microcomputer, you are responsible for the production, finance, health and well-being of Project Omega, the Earth's first deep space colony.

Painstakingly researched, Project Omega will provide much enjoyment and satisfaction as you overcome the frustrations and obstacles of taming an uncharted environment. The tape version is for one player; the disk version supports one or more, plus a special tournament option.

16K Tape...$14.95 32K Disk...$24.95



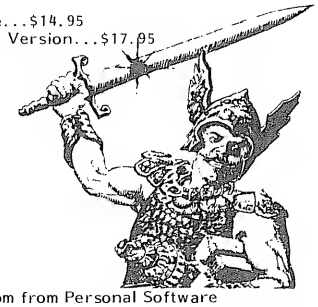
ATTACK FORCE!

By B. Hogue & J. Konyu from Big Five
Unlike the usual space "shoot em ups," your ship is not tied to the bottom of the screen. In Attack Force, you use the arrow keys to control both speed and direction as you maneuver all over the screen in search of the alien Ramships and Flagships. A realtime, machine language game with amazing graphics and sound.

You have to be quick to avoid the enemy ships that warp down on you, and the Flagships' lasers can fire in all directions -- even diagonally! And don't look away for an instant, because one of the alien spacecraft might be transformed into a mirror image of your own!

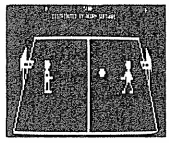
16K Tape...$14.95
32K Disk Version...$17.95

ZORK



By Infocom from Personal Software
In Zork, the Great Underground Empire, unearthly creatures guard 20 treasures. Bring all the treasures back to the trophy case and you can leave alive! You must pick your way through intricate mazes, collecting objects that may help or hinder you in your quest. But keep your wits about you, because in Zork, they take no prisoners!

TRS-80 or Apple II, 32K Disk...$39.95

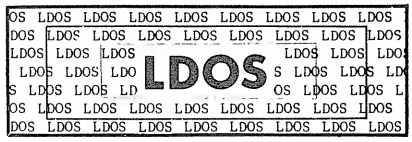


BASKET BALL

By John Allen from Acorn
You have to be fast to keep up with the action as you try to outscore your opponent in five minutes of one-on-one basketball. Compete against a friend or your computer.

Steal the ball, duck around your opponent and slant toward the basket for a lay up! The graphics are based on a 3-dimensional depiction of a basketball court, and ball dribbling sounds add to the realism.

Protected Tape...$14.95
Protected Disk...$20.95



JUST ANOTHER PRETTY DOS?

If new LDOS were just another disk operating system (DOS), we would not recommend it to you. However, two differences make this system unique and important: customer support and user benefits.

When you buy any DOS, you need service for programming assistance, updates and tips on how best to use its features. This is part of what you buy with LDOS. First, you will be supported by a toll-free phone line listed on your registration card. Second, you will be promptly notified of all updates and may send your original diskette to a service center for updating. You pay only the cost of return mail you can do it every week if you like. Third, a regular newsletter will inform you of any updates and provide tips on using some of LDOS's many special features.

With LDOS you get a well documented, thoroughly tested, and powerful DOS. The publisher is committed to a professionally written and detailed users' manual. Besides contracting with some of the best microcomputer systems houses for technical and customer support, a highly regarded technical writing firm is doing the manual.

The power of LDOS is its ease of operation, its independence of hardware configuration, and its device independence. You can make selected backups, chain together a series of programs and operations, and operate several different types of drives from the computer. LDOS has all the features of VTOS 4.0, fully implemented and working. There are hundreds of features which we do not have room to mention, but the best features are its ease of use for the new disk owner combined with its sheer power potential for the expert. We will gladly send you more information on LDOS if you just call our toll free number.

LDOS with Manual...$139

---- INTRODUCTORY FREE BONUS ! ----
(A $32.50 Value)

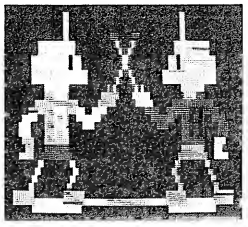
Purchase LDOS before March 31, 1981 and we will include a box of 10 diskettes FREE!

Visit Our New Store: W. Bell Plaza – 6600 Security Blvd · Baltimore, MD

TO ORDER CALL TOLL FREE 800 424-2738 For information Call (202) 337-4691

THE PROGRAM STORE Circle 17
4200 Wisconsin Avenue NW, Dept. KR8 Box 9609
Washington, D.C. 20016

MAIL ORDERS: Send check or M.O. for total purchase price, plus $1.00 postage & handling. D.C. residents, add 5% tax. Charge card customers: include all embossed information on card.



new

DUEL «N» DROIDS

By Leo Christopherson from Acorn
Your 'droid has already learned NIM, so now it's time to teach it how to wield a laser sword! Leo Christopherson, author of "Android NIM," "Dancing Demon" and other animations, has developed a new type of animation and high-quality sound in his latest work.

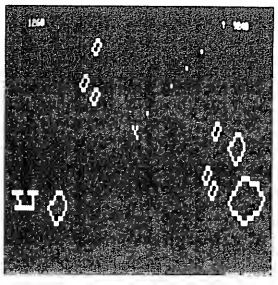
Your 'droid starts out as a lowly clown. You teach it how to use a laser sword by controlling its movements. After training it to be a "Grand Master," you enter the tournament against the program's skilled 'droid! Entertainment for all ages.

Protected Tape...$14.95
Protected Disk...$20.95

DEATH-MAZE 5000

from Med Systems
A new breed of adventuring! Venture through a graphically represented 3-D maze, with halls that could dead end -- or recede to infinity. Step through the doors or drop into the pits. Will you encounter monsters and mayhem, or will you be treated to useful objects and information? Will you ever get out alive?

16K TRS-80, 32K APPLE II...$12.95



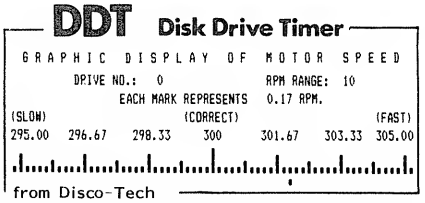
SUPER NOVA

By Bill Hague from Big Five
Asteroids surround your ship. You must shoot the asteroids, as well as any alien spaceships. Written in fast machine code, this game is GREAT!

You may encounter five different kinds of alien ships, including the very deadly flagship. You shoot from your ship's position, rotate it, use your thrusters to move -- if you are overwhelmed, you can even get away to hyperspace. Fast and exciting.

Tape....$14.95

DDT Disk Drive Timer



from Disco-Tech
Analyze and adjust your disk drive motor speed with a real-time graphic display. Manual details use for Radio Shack, Shugart, MPI, Pertec and Vista drives, and DDT can be used with any drive. All you need is DDT, two screwdrivers and five minutes.

Disk.....$19.95

SPACE WAR

By Device Oriented Games from Acorn
A two-player, real-time action game that lets each player control a spaceship with rotate, thrust, fire, and hyperspace. Five game options (including gravity) and three playing speeds. In fast machine language.

Tape...$9.95

GALACTIC TRILOGY

By Douglas Carlston
Take control of the Galactica as you navigate through an uncharted 3-dimensional universe. In "Galactic Empire," you attempt to unify a universe that is randomly created each time you play.

"Galactic Trader" pits your bartering skills against those of the other inhabitants as you try to accumulate riches and power. But watch out for the assassins and the energy cartel -- they're out to getcha!

Diplomacy and deviousness play equal parts in "Galactic Revolution." It's a game that combines tactics, social manipulation and Machiavellian ruthlessness. For more intrigue, this game allows more than one player. Sound effects.

Choose any game at $14.95 for TRS-80 16K on tape, $24.95 for Apple II 48K Disk.

To control the entire universe, get all three!

JET FIGHTER PILOT

from Instant Software
Launch one of several realistic jet fighters from an airport, or catapult from the deck of an aircraft carrier. Incredibly realistic simulation, right down to maintenance problems.

You will not only learn about the dynamics of flight, you'll discover the complex operation of modern military jet aircraft as you sit back and try to keep up with the constantly changing instrument panel display. Challenging and informative.

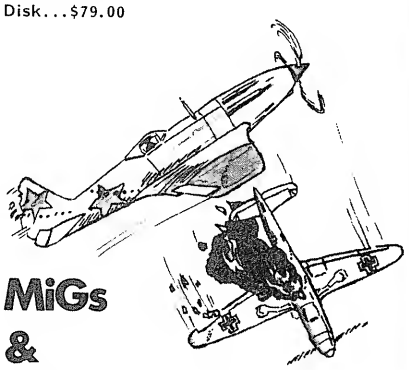
Cassette...$14.95

EDAS Editor/Assembler

By Roy Soltoff from MISOSYS
With EDAS, you are no longer tied to memory limitations while writing in assembly language. Now you can assemble directly from text stored on disk. Branching lets you test your program, then return directly to EDAS. Great for editing and debugging.

Other features include: global editing, upper/lower case support, block moves, plus availability of DOS commands within EDAS. It's the Editor/Assembler designed with the programmer in mind!

Disk...$79.00



MiGs & Messerschmidts

"It is the summer of 1941 and the Blitzkreig is smashing into the heart of Russia..."

This is how your instructions begin when you become the fighter squadron leader in "Migs & Messerschmitts", one of four exciting new Discovery Air Combat Simulations.

These World War II re-enactments are historically accurate -- they challenge you to learn the tactics used by the actual combatants! Written in machine language for fast response.

MIGs and Messerschmidts
RAF: The Battle of Britain
Jagdstaffel
Winged Samurai

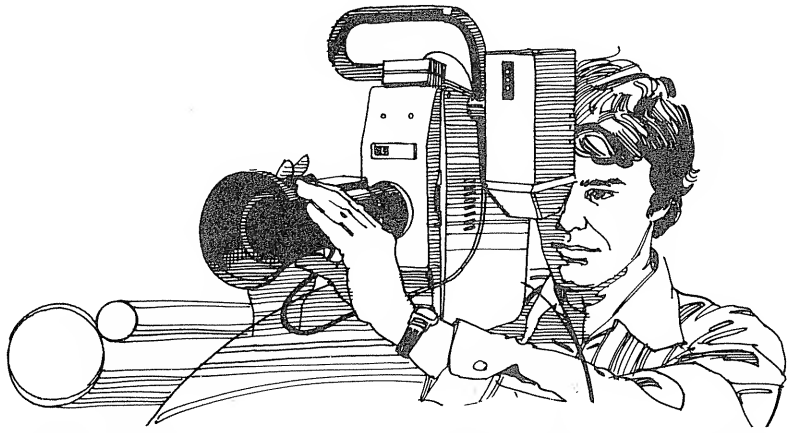
For TRS-80, Apple II, PET -- 16K...$19.95

PROGRAMS UNLIMITED...

...if you don't see the program you'd like, give us a call -- we probably have it!

NEWDOS+*.....99.95
NEWDOS/80*.....149.95 RADEX 10*.....99.95
Level III BASIC.49.95 CCA Data Mgr*....75.95
MMS FORTH /Man*.79.95 SYSTEM SAVERS....14.95
KeyEdit.....18.95 Tiny PASCAL*.....50.00
Acorn w/Sound (each): Tape 14.95 Disk 20.95
Pinball Pigskin StarTrek Ting-Tong
Invaders From Space Alien Invasion
Language Teacher (Fr, Span, Ital.)* 19.95 ea.
OSBORNE & ASSOC. Business Systems*: 25.00 ea.
General Ledger Accounts Payable
Accounts Receivable Payroll w/o cost acct.

* Disk



VIEW FROM THE TOP OF THE STACK

The 15th in a Series on Machine Language Tutorials.

In the last issue we promised to go into getting information between a USR routine and a Basic program. There are many ways to do this, but we will consider only two for now. First, we will pass information straight to the routine as the argument of a USR call. Second, we will pass arguments by putting them in a string and put the string in a buffer area we designate.

Each of these methods has some interesting uses. It does require PEEK and POKE though, so it is not applicable to the Model II unless you have modified your BASIC in Model II to include PEEK and POKE.

There are other methods for passing arguments that we will go into later in the series, but these will serve you well for now.

USR Arguments

Passing numbers or letters (as ASCII codes) through the USR call is easy but limiting. If we want to pass the number N to the routine, we have the Basic statement:

Y = USR(N)

The problem with this is that N can't be anything other than an integer. If it is, it is converted into an integer before it is passed to the routine. If the number is too big to be an integer an error results.

In order to get the value of the argument and make it available to the machine language program, we CALL 0A7FH in either Model I or III. This will place the value of N in the HL register pair as a two byte signed integer.

If we want to send a value back to the Basic program as the value to be assigned to Y, we load the value into the HL register pair as a two byte signed integer, then we JP 0A9AH in either Model I or III.

What if we needed to get more than one argument to the machine language routine? In that case, we can simply call the USR routine several times in succession. In the machine language routine, we would simply have a beginning section that would accept an argument, increment a count, and then return if the count wasn't yet high enough.

How do we get more than one argument back from the routine? There we have a stumper in this method. We can't simply return because when we start, we would have to have a routine to determine if we are entering to start processing or to get more returning arguments. If we put such a routine in front, we could key it with a unique integer argument and test for it to say start computation or go get the output values.

This kind of routine rapidly becomes the computer's answer to Rube Goldberg. Instead, we look for other ways to pass arguments.

Buffers

We can also play some tricks with VARPTR to place variables in known locations where our machine language program can access them. If you look in your reference manual, it tells you what information is located where VARPTR is pointing to. For a string, PEEK (VARPTR(A$)) is the length of the string, PEEK (VARPTR(A$)+1) is the LSB (least significant byte) of the string's address and PEEK (VARPTR(A$)+2) is the MSB (most significant byte) of the string's address.

If we know that we have a 20 character buffer for the machine language program located at memory location 7D00H (32000 decimal), we could define a string, say BF$ in our program. Once the string is defined, we then reassign it's memory location to correspond to the buffer like this:

```
POKE(VARPTR(BF$)),20
POKE(VARPTR(BF$)+1),0
POKE(VARPTR(BF$)+2),125
```

This puts the length of the string at 20 and reassigns the LSB of its address to 0 and the MSB to 125 (7DH).

Once the reassignment is complete, we can now use this string to pass information between the Basic program and the machine language program. This technique can also be used with numbers, but we have to do it differently.

With numbers, the actual number is stored at the location given by VARPTR. To access it then, it is easier to get the value of VARPTR to the machine language routine and let is use that as a buffer for the number.

Remember what VARPTR is. It's the address of a variable in memory. That is, it is a two byte number. In fact, we can pass it as a USR argument to our routine and then use it to access the variables where they are. In this way, if we wanted to use a whole double precision number (eight bytes), we could simply pass the VARPTR of the variable to our routine. When the address shows up in the HL register, we use that address to find the information we want.

The same technique will work with strings. Pass the VARPTR of the string and then access it from the machine language routine. That way is easier if you are not using the string as a buffer but want to modify it in place.

You do have to be careful when accessing Basic variables directly. You can easily make changes in the variable and then not have them reflected in the way Basic sees them if you forget to update the variable references.

As an example, let us say you have a

program that works as a text editor and you are using machine language routines for actual text insertion and deletion. If you make a string longer or shorter, you have to go back and change the VARPTR locations to make sure they agree with what you are actually returning to the program.

Try each of these ways to pass information to a USR routine until you are sure you can handle them. Then try the assignment.

Assignment

This issue's assignment is to write a machine language program to be used as a USR routine for a Basic program. Pass the routine the location of the first byte of an integer array (don't forget to DEFINT in the Basic program). Use the USR routine to sort the array into numerical order by doing a bubble sort.

A bubble sort is done by simply going through the array one element at a time, comparing one element with the next and exchanging them if they are not in correct order. This is repeated until a pass is made where there are no exchanges.

You can either pass the number of elements to be sorted to the USR routine by a USR call or let the routine find the number of elements in the array by looking from the VARPTR point. Note that VARPTR returns the address of the first element of the array. For an integer array, each element is two bytes long. Just before the array, the dimension of the array is stored, two bytes per dimension.

If this seems a little hard to you, or you want something else to do to while away your time, they try writing a routine that will pass a string to the routine through a buffer. Sort the letters in the string into order and return the string in the buffer using the same technique described above.

We'll give our versions of these routines next issue. In the next column, we'll go over the sorting routine in detail and then start moving large amounts of data in memory. For example, moving a whole screen to a temporary holding place and replacing it with another screen. All that next time.

For Your Interest

We'd like to see your solutions to problems suggested in View from the Top of the Stack if you have interesting ones. Send a tape or disk with a description of your program and operating instructions to: View, 80-U.S. Journal, 3838 South Warner St., Tacoma, WA 98409. Include the source code for your program, fully documented, instructions for use, and comments about what kind of system you are using and what kind of assembler. Particularly interesting solutions may be published. Be sure to include permission to print the solution with your signature. Programs submitted to this column will not be returned unless accompanied by a self-addressed, stamped envelope.

Last Issue's Problem

The listing gives one short solution to last issue's problem. This problem has run on an old TRS-80, as well as a brand new one. ●

```
                00100 ;*****************************************************************
                00110 ;         VIEW FROM THE TOP OF THE STACK
                00120 ;
                00130 ;
                00140 ;         DEMONSTRATION OF ROM ROUTINES
                00150 ;         FOR INPUT AND OUTPUT OF INFORMATION
                00160 ;
                00170 ;*****************************************************************
                00180 ;
                00190 ;         INITIALIZE ROUTINE NAMES
                00200 ;
1A19            00210 LII       EQU     1A19H           ;LEVEL II ENTRY POINT
402D            00220 DOS       EQU     402DH           ;DOS ENTRY POINT
01C9            00230 CLS       EQU     01C9H           ;CLEAR SCREEN ROUTINE
0049            00240 GETCH     EQU     0049H           ;CHARACTER INPUT
0033            00250 PUTCH     EQU     0033H           ;DISPLAY CHARACTER
00F0            00260 BSIZE     EQU     0F0H            ;BUFFER SIZE
                00270 ;
                00280 ;         MAIN PROGRAM
                00290 ;
7000            00300           ORG     7000H           ;WHERE TO PUT IT
00F0            00310 BUFFER    DEFS    BSIZE           ;INPUT BUFFER
70F0 CDC901     00320 START     CALL    CLS             ;CLEAR THE SCREEN
70F3 214C71     00330           LD      HL,PMPT1        ;GET THE 1ST PROMPT
70F6 CD1E71     00340           CALL    PUTLN           ;DISPLAY IT
70F9 CD2771     00350           CALL    GETLN           ;GET A LINE
70FC 210070     00360           LD      HL,BUFFER       ;GET BUFFER
70FF CD1E71     00370           CALL    PUTLN           ;DISPLAY IT
7102 3E0D       00380           LD      A,0DH           ;GET A <CR>
7104 CD3300     00390           CALL    PUTCH           ;DISPLAY IT
7107 216071     00400           LD      HL,PMPT2        ;GET 2ND PROMPT
710A CD1E71     00410           CALL    PUTLN           ;DISPLAY IT
                00420 ;
                00430 ;         SINGLE CHARACTER INPUT FOR SELECTION
```

```
                00440 ;
710D 060D       00450 CHLOOP  LD      B,0DH           ;<ENTER> KEY VALUE
710F CD4900     00460         CALL    GETCH           ;GET A CHARACTER
7112 B8         00470         CP      B               ;IS IT <ENTER>?
7113 CAF070     00480         JP      Z,START         ;IF YES, THEN START
7116 061F       00490         LD      B,31            ;<CLEAR> KEY VALUE
7118 B8         00500         CP      B               ;IS CHAR <CLEAR>?
7119 CA2D40     00510         JP      Z,DOS           ;IF YES, THEN END
711C 18EF       00520         JR      CHLOOP          ;OTHERWISE LOOK AGAIN
                00530 ;
                00540 ;       I/O ROUTINES
                00550 ;
711E 7E         00560 PUTLN   LD      A,(HL)          ;GET CHARACTER
711F B7         00570         OR      A               ;IS IT ZERO
7120 C8         00580         RET     Z               ;IF YES, THEN DONE
7121 CD3300     00590         CALL    PUTCH           ;DISPLAY IT
7124 23         00600         INC     HL              ;NEXT CHAR
7125 18F7       00610         JR      PUTLN           ;LOOP
7127 210070     00620 GETLN   LD      HL,BUFFER       ;GET BUFFER
712A 0E01       00630         LD      C,1             ;SET COUNTER
712C 3EEE       00640         LD      A,BSIZE-2       ;GET BUFFER SIZE
712E 08         00650 BLOOP   EX      AF,AF'          ;STORE IN ALT REG SET
712F 0D         00660         DEC     C               ;DROP COUNTER 1
7130 CD4900     00670 ILOOP   CALL    GETCH           ;GET A CHARACTER
7133 060D       00680         LD      B,0DH           ;<CR>
7135 B8         00690         CP      B               ;IS IT <ENTER>?
7136 CA4671     00700         JP      Z,DONE          ;IF YES THEN FINISH
7139 0C         00710         INC     C               ;INCREMENT COUNTER
713A 08         00720         EX      AF,AF'          ;GET MAX COUNT
713B B9         00730         CP      C               ;COMPARE TO NUMBER INPUT
713C 38F0       00740         JR      C,BLOOP         ;IF #CH>BSIZE THEN LOOP
713E 08         00750         EX      AF,AF'          ;GET CHAR BACK
713F 77         00760         LD      (HL),A          ;OTHERWISE, PUT IN BUFFER
7140 CD3300     00770         CALL    PUTCH           ;DISPLAY IT
7143 23         00780         INC     HL              ;NEXT POSITION
7144 18EA       00790         JR      ILOOP           ;LOOP
7146 3600       00800 DONE    LD      (HL),00H        ;END WITH 00H
7148 CD3300     00810         CALL    PUTCH           ;DISPLAY THE <ENTER>
714B C9         00820         RET
                00830 ;
                00840 ;       PROMPT DEFINITIONS
                00850 ;
714C 57         00860 PMPT1   DEFM    'WHAT IS YOUR NAME?
715F 00         00870         DEFB    00H
7160 50         00880 PMPT2   DEFM    'PRESS <ENTER> TO REPEAT'
7177 0A         00890         DEFB    0AH
7178 4F         00900         DEFM    'OR <CLEAR> TO END'
7189 00         00910         DEFB    00H
70F0            00920         END     START
00000 TOTAL ERRORS
BLOOP  712E 00650   00740
BSIZE  00F0 00260   00310 00640
BUFFER 7000 00310   00360 00620
CHLOOP 710D 00450   00520
CLS    01C9 00230   00320
DONE   7146 00800   00700
DOS    402D 00220   00510
GETCH  0049 00240   00460 00670
GETLN  7127 00620   00350
ILOOP  7130 00670   00790
LII    1A19 00210
PMPT1  714C 00860   00330
PMPT2  7160 00880   00400
PUTCH  0033 00250   00390 00590 00770 00810
PUTLN  711E 00560   00340 00370 00410 00610
START  70F0 00320   00480 00920
```

EXCITING NEWS FOR INVESTORS WITH TRS-80™ 32K DISK-BASED SYSTEMS!

Standard & Poor's proudly announces STOCKPAK, a unique software and data system to help you meet your investment goals like a Wall Street professional.

STOCKPAK not only delivers a "stand-alone" Portfolio Management System but also gives you the software for Standard & Poor's monthly Common Stock Data Service (available to TRS-80 owners on a subscription basis). With STOCKPAK and the Data Service you command one of the most powerful and versatile investment tools available.

Here's How STOCKPAK Will Help You:

A 900 COMPANY DATA BASE SERVICE

Monthly Data Service subscribers receive a diskette containing 30 vital financial items on 900 of the most widely traded stocks (S&P "500" and 400 NYSE, ASE and OTC issues). Accompanying this monthly diskette is an Investor's Newsletter highlighting important financial news and investment strategies, with suggestions for maximizing the usefulness of the system.

STOCKPAK SELECTION SYSTEM

The heart of STOCKPAK is a powerful, analytical stock selection tool which enables investors to choose stocks which meet their investment criteria. For example, you may wish to select only those oil and gas stocks with price/earnings ratios of less than 7 and yields of 6% or more. Once a group of stocks has been selected, you can store it as a separate data file for continuing use.

REPORT WRITER

You can define the report formats you would like to see on those stocks meeting your investment objectives. Hundreds of calculations and ratios that you define can be sorted, averaged or totalled, and displayed on video screen or optional printer.

PORTFOLIO MANAGEMENT SYSTEM

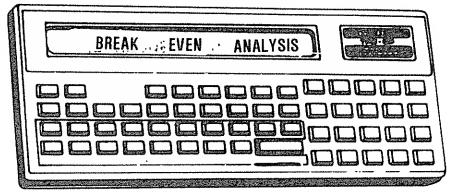
Now you can effectively evaluate and manage your own stock portfolio of up to 100 securities with as many as 30 transactions for each. You can record "buy" and "sell" transactions, price and dividend information and stock splits for instant retrieval, for record keeping and tax purposes. You can measure actual performance or create hypothetical situations to help you make "buy" or "sell" decisions.

HOW TO ORDER STOCKPAK

STOCKPAK is designed exclusively for TRS-80 users with 32K business systems with two mini-disk drives. You can obtain the basic software and sample Data Base, plus a comprehensive User's Manual from your local Radio Shack Store for only $49.95. The STOCKPAK Monthly Data Updating Service can be ordered directly from Standard & Poor's for $200 annually, or from the order form provided in the basic package you purchase from Radio Shack.

Standard & Poor's Corporation
25 BROADWAY, NEW YORK, NY 10004 (212) 248-3994/3374

MANAGEMENT



and the Pocket Computer

John D Farrell
Tacoma, Washington

Ever since the electronic calculator has been made available at a reasonable cost my wife has been accusing me of being a "collector". When Radio Shack brought out their new hand held computer I naturally couldn't wait to get my hands on one. I wanted to discover what practical uses could be made for management decisions on a day to day basis. The only problem was that I knew absolutely nothing about programming! This article will carry you with me from my first attempt at addition and subtraction through my first usable program.

Step one: When all else fails, read the directions. I got no further than the introduction to the manual that stated, "If you are looking for a 'lead-em by the hand manual, this is not it'." They were right, I went to Radio Shack and purchased a couple of books for the beginning programmer. For the next few weeks I felt like a student in a foreign country having to translate all my studies from one language to another.

My initial impressions after learning how to calculate with all the formulae I had not seen since high school, was that the pocket computer would be valuable, perhaps a necessity, for the engineer or a like profession. What possible purpose could it have for someone in the restaurant business?

A few more weeks went by as I struggled with heretofor very simple words, such as "LET", "INPUT", "PRINT", "CLEAR", etc. and I began to realize that the computer could indeed do some things that would be useful. Since management must have a successful game plan to run, a good deal of management time is spent forecasting results based upon both fixed and variable costs. I thought that a break-even analysis would be a good test to see if the pocket computer could become useful to me.

As a consultant, I found that many restaurant owners simply had no idea how their total cost of sales percentage was affected by their sales ratios of food, beverage, and other income categories. Most were also not aware of how many income dollars were necessary to support their fixed expenses after all the variable costs were paid. Fixed expenses are those that do not change with fluctuations of dollar volume; variable costs are those that are directly affected by sales.

What follows then, is a very simple program written by a novice programmer to help forecast what sales revenues are necessary to return a satisfactory profit to a restaurant owner. I have also included some line by line instructions for those of you who may read this and are novices both to the programming or the restaurant fields.

1. Lines 10 and 20 use the Pause statement to define the program and give credit to the author.

2. Lines 30, 180 and 310 define the different functions of the program. Line 30 also clears all memory locations.

3. Lines 40 through 100 put projected sales and budgeted percent of costs in the A through M locations of the memory.

4. Lines 110, 120 and 130 are simple formulae that add the sales forecast and calculate both the amount of sales costs and the percent of cost to sales.

5. Lines 140, 150 and 160 print the results.

6. Lines 135, 155, 295, 425, 445, 475 and 485 tell the computer to print the results of the formulae in the appropriate format of dollars and cents or percentages to the nearest tenth. This is an example of the subroutine capability of the computer.

At this point in the program the only information needed for the break-even analysis is the percent forecasted for cost of sales. Therefore, all memory locations are cleared by the Clear statement on line 190 to make more memory locations available for the rest of the program.

7. Lines 190 through 280 put all the projected variable cost of sales into memory locations A through J.

8. Line 290 again uses a simple formula to add the variable percentages.

9. Line 300 prints the total variable costs as a percentage of sales.

10. Lines 320 through 410 input the fixed expenses in memory locations L through U.

11. Line 420 again simply adds the fixed expenses.

12. Line 430 prints that result.

13. Line 440 is the break-even formula.

14. Line 450 prints the amount of sales dollars needed in order for the restaurant to break even.

15. Lines 460 through 490 determine both profit dollars and profit percent in relation to total sales. The program user can put in as many different sales projections as he needs and the program will figure the profit or loss.

My conclusions are that the pocket computer can indeed be a very useful tool for helping the manager in budgeting, forecasting and a variety of planning functions. I look forward to exploring the science of computers as I am sure that the manager of tomorrow will need this technology in an increasingly competitive and complicated environment.

```
10: PAUSE "BREAK EVEN ANALYSIS"
20: PAUSE "BY JOHN FARRELL 1/1/81"
30: CLEAR: PAUSE "COST OF SALES:"
40: INPUT "FOOD SALES = ";A:INPUT "% OF COST = ";B
50: INPUT "LIQUOR SALES = ";C:INPUT "% OF COST = ";D
60: INPUT "BEER SALES = ";E:INPUT "% OF COST = ";F
70: INPUT "WINE SALES = ";G:INPUT "% OF COST = ";H
80: INPUT "CIG./CANDY SALES = ";I:INPUT "% OF COST = ";J
90: INPUT "POOL/GAMES = ";K:INPUT "% OF COST = ";L
100: INPUT "OTHER SALES = ";M:INPUT "% OF COST = ";N
110: O=A+C+E+G+I+K+M
120: P=((A*B)+(C*D)+(E*F)+(G*H)+(I*J)+(K*L)+(M*N))*.01
130: Q=(P/O)*100
135: GOSUB 600
140: PRINT "TOTAL SALES = ";O
150: PRINT "COST OF SALES = ";P
155: GOSUB 700
160: PRINT "COST OF SALES = ";Q
180: PAUSE "VARIABLE EXPENSES"
190: CLEAR: INPUT "% COST OF SALES = ";A
200: INPUT "% REPAIR/MAINT. = ";B
210: INPUT "% ADVERTISING = ";C
220: INPUT "% BAD DEBTS = ";D
230: INPUT "% BANK DISCOUNTS = ";E
240: INPUT "% LAUNDRY = ";F
250: INPUT "% CHEM/PAPER = ";G
260: INPUT "% PAYROLL = ";H
270: INPUT "% RENT = ";I
280: INPUT "% OTHER = ";J
290: K=A+B+C+D+E+F+G+H+I+J
295: GOSUB 700
300: PRINT "VARIABLE & = ";K
310: PAUSE "FIXED EXPENSES:"
320: INPUT "EMP. INSURANCE = ";L
330: INPUT "ACCOUNTING = ";M
340: INPUT "DEPRECIATION = ";N
350: INPUT "EQUIP. LEASE = ";O
360: INPUT "INSURANCE = ";P
370: INPUT "OVER/SHORT = ";Q
380: INPUT "MUSIC/BAND = ";R
390: INPUT "OFFICE EXPENSE = ";S
400: INPUT "UTILITIES = ";T
410: INPUT "OTHER EXPENSES = ";U
420: V=(L+M+N+O+P+Q+R+S+T+U)
425: GOSUB 600
430: PRINT "FIXED EXP. = ";V
440: W=V/(100.00-K)*100
445: GOSUB 600
450: PRINT "BREAK EVEN = ";W
460: INPUT "PROJECTED SALES = ";X
470: Y=((X-W)*(100.00-K))*.01:Z=(Y/X)*100
475: GOSUB 600
480: PRINT "PROFIT = ";Y
485: GOSUB 700
490: PRINT "PERCENT = ";Z;"%":GOTO 460
600: USING "#######.##":RETURN
700: USING "####.#":RETURN
```

DOSPLUS 3.1

1.5 MEG on MODEL 1

Increase Your Disk Storage!!!

Package 1 - 1.5 Megabytes for only $1,540.00
2: Model 160-2 AEROCOMP 80-track double headed drives
1: PERCOM DOUBLER
1: DOSPLUS 3.1D
1: 2 Drive Cable (for the AEROCOMP drives)

Package 2 - Upgrade your Model I to a Model III for only $320.00
1: PERCOM DOUBLER
1: DOSPLUS 3.1D

Package 3 - Add on disk storage and go double density for only $1,040.00
2: Model 40-1 AEROCOMP 40-track "flippy" drives
1: PERCOM DOUBLER
1: DOSPLUS 3.1D
1: 4 drive cable (for the AEROCOMP drives)

Package 4 - Add an 80-track drive and double density for only $800.00
1: Model 80-1 AEROCOMP 80-track "flippy" drive
1: PERCOM DOUBLER
1: DOSPLUS 3.1D
1: 3 drive cable (for the AEROCOMP drive)

Remember, ONE 80-track drive, running double density, will give you as much storage as FOUR 35-track drives running single density.

If interested, call or write us at the address below, and let us banish your disk storage problems forever!

MICRO SYSTEMS SOFTWARE INC.
5846 Funston Street, Hollywood, FL 33023
(305) 983-3390

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

PHONE _____

ACCOUNT # _____

MC ☐ VISA ☐ EXP. DATE _____

PROGRAM NAME _____

QUANTITY _____

AEROCOMP DISC DRIVES

FOR TRS — 80*

• MODEL 80-1 DISC DRIVE $449.95 ea.
Single-sided, "flippy", 96TPI (80 track; single density unformated 250K bytes/side; double density unformated 500K bytes/side).

• MODEL DISC DRIVE 160-2 $599.95
Double-sided, 96TPI (160 track/80 per side; single density unformated 500K bytes; double density unformated 1 megabyte).

All models are capable of single or double density and are complete with power supply and silver enclosure. Send for information on AEROCOMP 2-and 3-drive systems available in 40 and 80 track.

The Doubler™: Percom's new proprietary double-density adapter for the TRS-80* computer.

- **Increase formatted storage capacity of your minidiskettes from 1 1/2 to almost 4 times.**
- **Use with standard 5-inch drives rated for double-density operation.**
- **The DOUBLER™ reads, writes and formats either signle - or double density disks. Only $219.95.**

NEW FROM MICRO-SYSTEMS!!!

Micro-Systems Software Inc. now has double density software available for TRS-80* Model I's that are equipped with the Percom Doubler.+

First is a disk editor called "Disk Zap 2.3". This editor will work either single or double density diskettes. It is track and sector oriented, and offers total access to all parts of the disk. It has the ability to format and backup diskettes as well as editing them.

Second is our new double density DOS. DOSPLUS 3.1D, like our regular DOS, will run 35-80 track drives; but offers the increased disk storage of double density.

Disk Zap is $19.95 and DOSPLUS 3.1 or 3.1D is $99.95. To order, call or write us at the address below. Master Card and Visa welcome. Orders accompanied by a personal check will be shipped when the check clears the bank.

master charge | VISA

MICRO SYSTEMS SOFTWARE INC.
Specializing in the Tandy Line

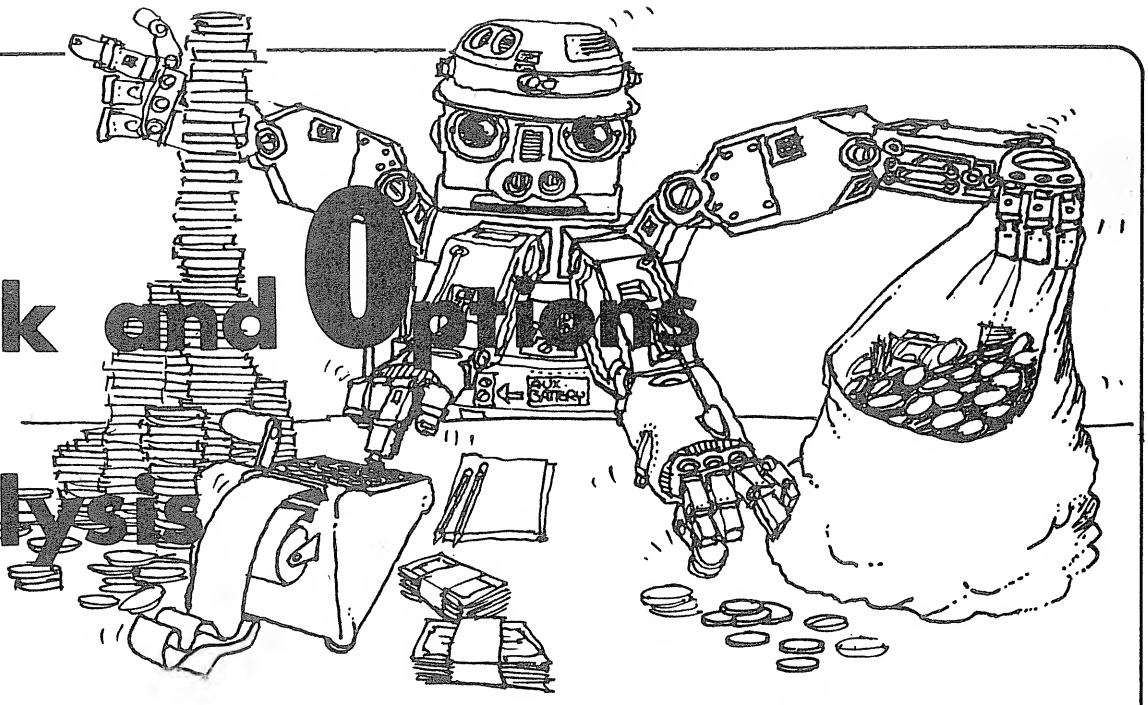
(305) 983-3390

5846 Funston Street
Hollywood, FL 33023

* TRS - 80 is a trademark of Tandy Corp.
+ Doubler is a trademark of Percom Data Corp
Now available in Spanish

Circle 85

Stock and Options Analysis



Keep the data you need to make timely investment decisions at your fingertips with this incredibly powerful investment tool. Considerable effort has gone into methods of tilting the odds in the investment game. Out of this has come the discovery that the strategy of hedging listed options against common stocks can tilt the odds drastically. In fact, it can be *more conservative* and *more consistently profitable* than the simple buying and selling of stock.

The four programs in this package are designed to be used in the real world, and include the effects of commissions, margin interest and dividends, where applicable. Possible investment attitudes, the listed option markets, puts and calls and option strategies are covered in extensive documentation.

The **Option** program presents important indices of both opening and closing call option transactions. The manual includes sample runs illustrating combination strategies with covered and uncovered calls, and covered and uncovered straddles recieve detailed treatment.

The **Opgraph** program presents a graph or a table, as the user chooses, of profit from any combination of six basic positions: long or short a stock, long or short a call and long or short a put. Sample runs are presented which cover hedging with calls, out-of-the-money hedges and in-the-money hedges.

Newprem enables the user to predict the future premiums of an option at whatever time and future stock price the user selects. This method requires the establishment of a data base of historical option premiums in whatever detail the user desires.

Finally, **Portval** enables the user to determine on an item by item basis, the cost, current value per share, total current value and capital gain of a portfolio consisting of long and short stock, and long and short option positions. This program assists the user in keeping a readily available and easily updatable record of his portfolio and, at the same time, assists him in measuring his progress towards financial success.

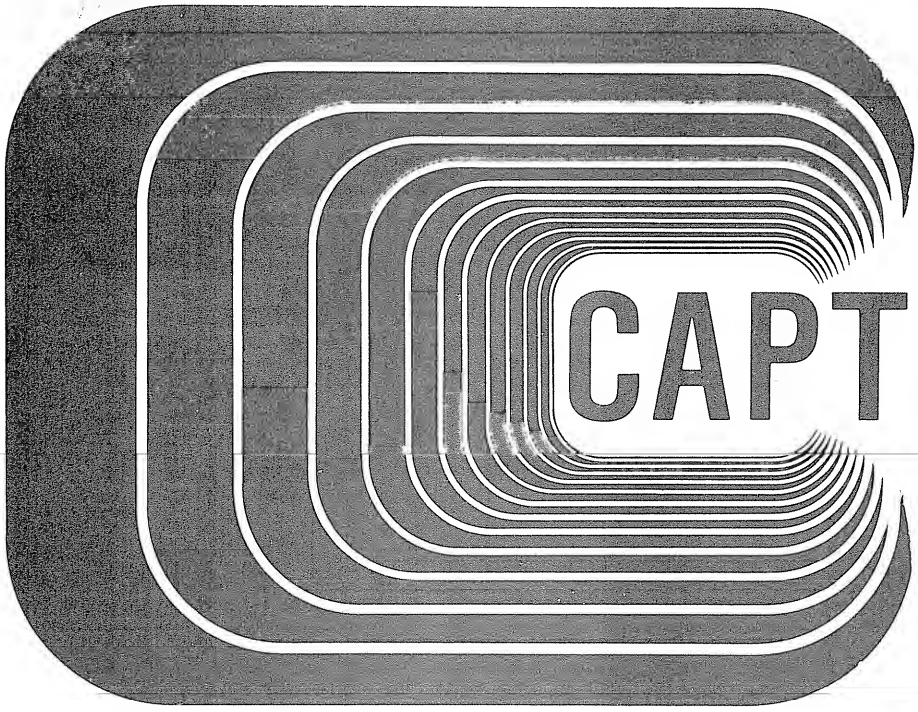
In order for an investor to continually improve his performance it is necessary for him to refer to past performance; this requires useful records. Finally, he should constantly be evaluating his performances to assure himself he is playing the right game.

The **Stock and Options Trading Analysis** package is available for the 16K TRS-80 Level II on cassette (CS-3306) and disk (CS-3801) for $99.95. Creative Computing Software should be available at your local computer store. If your favorite retailer does not stock the software you need, have him call our retail marketing department at the number below. Or you can order directly from Creative Computing Software, Dept USLG; P.O. Box 789-M, Morristown, NJ 07960. Visa, MasterCard, or American Express are also welcome. For faster service, call in your bank card order toll free to 800-631-8112. In NJ call 201-540-0445.

TRS-80 Professional Software

creative computing software

Circle 22



CAPTAIN 80™

Bob Liddil

Here's Captain 80 disguised as the Software Secret Agent, Sam Spade hat pulled low over my eyes and trenchcoat collar turned up against the wind. The mirror-shades over my eyes reflect a pile of software for review roughly the size of a small tractor trailer (10-4). They are products of the great, near great, good and good grief!

Scott Adams' Adventure International, (those wonderful folks who brought new meaning to the term "Instant Insanity") has, over the last twelvemonth, expanded their line to include many fine units of non-adventure software. Many of the country's finest authors have teamed up with Scott to produce what can only be described as great computer programs.

Strip Concentration and Dice, (rated X for no apparent reason), is an example of a company that does not take itself too seriously. The *Concentration* segment is a spoof of the old TV game show of the same name. It features hats, coats, shoes and the like. I'll leave the rest to your imagination. The *Dice* segment could just as easily be used for computerized Dungeons and Dragons if you could play that game to the monotone strains of "Let Me Entertain You". The game is a great spoof of *Interlude*, though at no time does it ever get even close to explicit. Scott has always had a great sense of humor and this program is funny.

I find a few faults with it though. When Agent 442432 and I tried to play test it, we discovered that there were no provisions for shoe-telephones, sword-belts, hatband-guns or utility girdles. Now what's a secret agent to do under circumstances like that?

Tunnels of Fahad is a microization of the popular arcade game *Head On*. In *Tunnels*, the computerized protagonist is a mummy who is trying to get you before you get the gold, which is laid out around a series of tracks. You have to get as much of the gold as you can in order to score before the inevitable time the mummy locks in on you and nails you. *Tunnels* scored an unbelievable triple star among the kid reviewers polled and is consistently requested by visitors to the Software Secret Laboratory. The author and publisher are to be congratulated for a fine product.

"You know why you plunked down all that money for a computer don't you? It was so you could play Poker and still stay home with the kids."

Frogs! How's that for the name of a prolific piece of business software? And the business, of course, is the catching and munching out on, flies. Picture if you will a giant computerized bullfrog, sitting on a log floating placidly in a graphics marsh. Above him flits flies. As the flies fly about, he eyes them hungrily. Here's where you come in. By manipulating various and assorted keys, you position the frog so that he is in a most favorable striking pose. Then, out shoots a zig-zagged, nanosecond fast tongue and *blip!* One less fly. Our frog then chews his munchie and returns to the game. This may seem frivolous, but wait. It is very serious. How is the frog going to survive without your help? And to add insult to injury, his tongue gets shorter each time he goes for a fly. Sound and near animation quality graphics give this program a star and a half. As tested on the Model III, it is highly requested and often played.

You know why you plunked down all that money for a computer don't you? It was so you could play *Poker* and still stay home with the kids. (*Poker* makes a great babysitter when you are not home with the kids too). Al's version of *Poker* sort of

reminds you of Doc and Kitty in the Longbranch Saloon. It lets you see your hand but not the others in the game. It plays a good game of draw and allows a good leeway for bluffing and strategy. The only reason it didn't draw a double star is due to previous utter burnout on card games experienced by yours truly. The star it did get is due largely to the snappy, entertaining style in which it is presented. I liked it and play it moderately, though right now I owe it three thousand dollars, so I'm laying low.

Slag takes some getting used to. It is a game of world politics and nuclear warfare that has some complication and sophistication, while appealing also to those of us who must have pictures. Industries are built, as well as nuclear arsenals. When war finally breaks out, it's interactive! The players take turns bombing each other's submarines, blasting missles out of the sky and eventually wiping one another out. But there are some economic simulations built in as well, where you must make money decisions that can affect the outcome of the game. *Slag* is not for the pacifist or even for everyone. It is well written though, and gets a well deserved star for a purely imaginitive treatment of a difficult subject.

There are other fine programs in Scott's new line but we will save them for another time.

There is a little company called *Computer Power and Light*, out of Northridge California, which has two of the tightest, hottest arcade style programs presented to date. They are *Spaceball* and *Galactic Fighter*. *Fighter* is an exquisitely executed version of Galaxians, an Invaders style game in which individual units of the enemy seek you out. It is fast-paced, perfectly animated and with sound effects that are well timed and effective. The second, *Spaceball* is like the first in quality. It is a version of Breakout worthy of any coin-op in any arcade.

It is rumored, as of this writing, that these two top quality programs are going to be marketed through Instant Software. I hope that *Computer Power & Light* will consider that since we have enough mega-giants in the industry as it is.

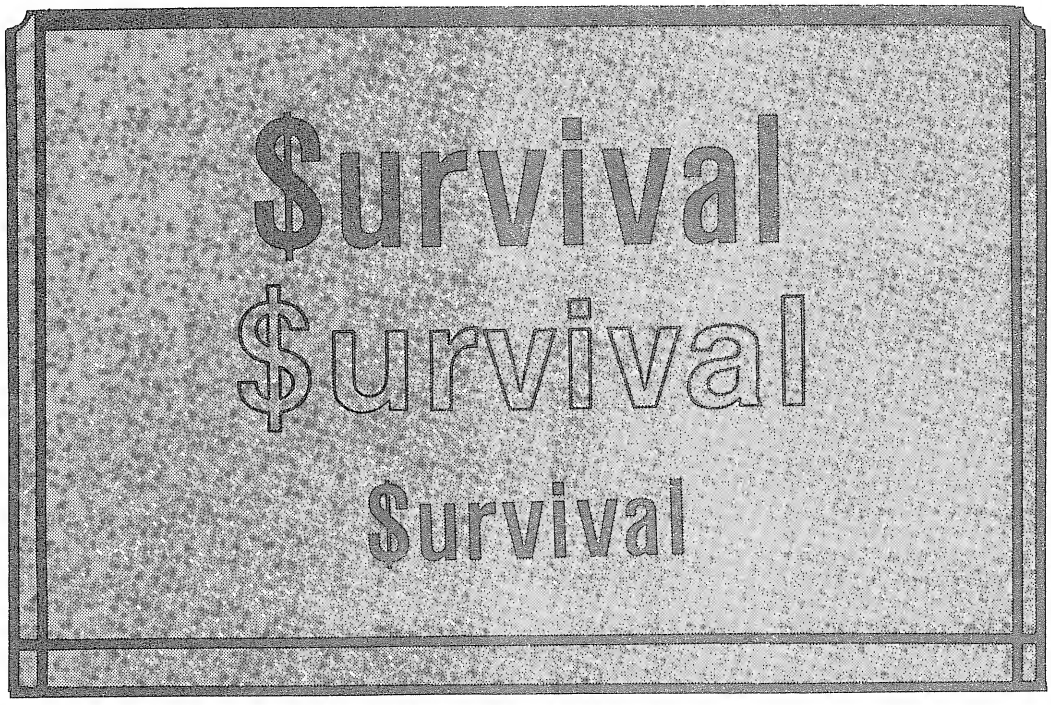
There are many young faces in the ranks of selling programmers. Some of them I know personally, some I have not met. All of them have one thing in common, degrees of vulnerability in the marketplace. There are many sharks out there waiting to gobble up naive programmers who don't or can't research companies they submit to.

For those youngsters (or oldsters for that matter) who actually survive long enough to collect on their efforts, the rewards presented by honest, legitimate vendors, in many cases aren't bragable. What's the answer?

For Greg Hassett, age 14, it was *Adventureworld*, his own company, run with the help of his parents. For Roy Neiderhoffer, age 15, it was *Software Innovations*, also his own company. For Charles Forsythe, age 15, newcomer, it was an agent and eventually the placement of a TRS-80 Color Invaders with Scott Adams. Success is possible.

For the Software Secret Agent, the stack of material is growing smaller. Only *Ultra-DOS*, Vern Hester's hand built disk operating system remains. Vern scratch built the BASIC for Ultra-DOS and added many enhancements to the already familiar format offered to us by NEWDOS. For the DOS-idiot like myself, *Ultra-DOS* was a true blessing. It has replaced for now, at least, both TRSDOS and other highly complex systems, as our lab-DOS. Now rumor has it that Vern, whose talents are limitless, and Kim Watt, who's powerful and usetul *Super Utility* caused so much controversey due to its ability to copy **anything**, have decided to collaborate on a Super DOS which is reputed to have drawn the interest of many of the giants of the industry. For those two working together is an event worthy of applause. In any case, at $119.95, *Ultra-DOS* is a byte in the billfold, but it does deliver a dollar to DOS even value.

Now from Software (and hardware) Secret Headquarters, this is your Secret Agent, signing off. ●



This Program Works Without Modification on Models I and III

Spencer Hall

Here is a program which makes frequent and effective use of the Z-Subs which were presented in the last issue of 80-U.S. Journal. We'll comment on those Z-Subs later. Right now I'd like to say that this program exemplifies my conviction that software for a personal computer should meet some personal need of the owner, whether it be for business or pleasure, and not just be something to watch because it is impressive. The general public today is continually amazed by the things minicomputers and microcomputer controlled devices can do. "Will wonders never cease?" It's what I call the "GEE WHIZ" reaction, but I give it about five years before "GEE WHIZ" will turn to "SO WHAT". When that happens it will be programs like $URVIVAL which will justify the cost of a personal computer.

So what exactly is $URVIVAL? No, it isn't a game...except to the extent that living within one's means in this modern world is a game. Actually, it's a personal finance program. But, before you stifle that yawn, let me assure that it isn't another of those orderly concoctions that post or balance your checkbook, record your expenses, income, or whatever.

For the average person on an average income (whatever that is!), checkbooks are easy to maintain by hand. What's more, tracking my budget, in the sense of making a running record of expenses in each period and comparing them with a spending plan, would be a waste of time...on or off my TRS-80. The unexpected is always happening.

If you live as I do, from one payday to the next, then $URVIVAL is for you. Even if your financial resources vary from month to month...as a matter of fact, especially if they do, $URVIVAL will help you manage better.

Figure 1 is a screen print of the job catalog you will see when you RUN $URVIVAL. If this is your first use, you will call job 1 (program lines 100-160) and write your budget. There will be cues calling for the account names and the amount you hope to (or must!) pay. You can have up to sixteen accounts. If you try to enter more, the program will cut you off (line 155). When your accounts are complete, answer the next request for an account name with the word NO. Now select job 2 which is a list of your accounts and their sum. Now the fun really begins. You're ready to run job 6...the moment of truth. Here's the idea. You start a period, whether it's a week, a fortnight or a month or whatever, with knowledge of what cash recources you have available, probably a paycheck. You subtract your expected expenses from your total resources then decide how frequently you want to give yourself an allowance to live on out of the money that remains. This is usually once a week...And there may be four or five "allowance days" in the month, depending on how the days fall and on what day of the week you like to draw your allowance. You divide the remaining money by the number of planned allowances and this tells you how much you can draw each time. If you want to save something, you treat that as another budget item, deducting it from resources before splitting up the remainder. Job 6 does all this for you automatically. Figure 2 is a fictional budget display generated by job 6.

Once around, this isn't a big deal to do on paper. But suppose you realize that you just can't live on that allowance which the program says you can have? How are you going to make up the deficit? You can omit some payments, of course, but wouldn't it be neat to specify how big an allowance you want and then go back and pick out certain bills to reduce in proportion to their size to make the allowance fit. This way you are splitting the available money to give each creditor the maximum possible payment...and you survive! Quite a hassle on paper but just a few keystrokes with job 7 of $URVIVAL. On this call you will be shown your budget and given a chance to designate, by number, up to eight of your accounts. The program will then reduce payments for these accounts, each in proportion to its size, to leave you exactly the allowance you have requested. When you have entered all the account numbers to adjust, you answer NO and you automatically get job 6 with the new allowance you asked for...and the reduced payments.

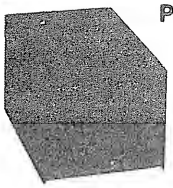
(Continued on Page 42)

Toll-free order no.:
1-800-527-4196

LOOK!!

TRS-80*Owners Save on Equipment & Software!

Percom Mini-Disk Systems



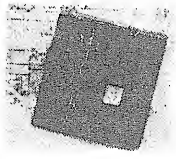
These Percom mini-disk systems store more data, are more reliable. Access times are fastest possible with your Expansion Interface. Heavy duty power supplies run cooler, last longer. Low noise three-wire ac power cord is safer. Enclosures are finished in compatible silver enamel. Prices:

TFD-100™ (40-track, 102 Kbytes/side)
- One-Drive Add-On ..... $ 349.95
- Two-Drive Add-On 699.45
- Three-Drive Add-On 1049.95

TFD-200™ (77-track, 197 Kbytes)
- One-Drive Add-On 634.95
- Two-Drive Add-On 1268.95
- Three-Drive Add-On 1903.95

Price includes Percom upgrade PATCHPAK™ program.

DATA SEPARATOR™



This PC board plug-in adapter for the TRS-80" virtually eliminates data read errors (CRC error — Track locked out!) which occur on high density inner disk tracks, a problem that has plagued TRS-80* systems. The Percom Data Separator™ is installed in the Expansion Interface without modifying the host system. Caution: Opening the TRS-80* Expansion Interface may void the limited 90-day warranty: $29 95.

Percom OS-80™

An advanced easy-to-use disk operating system that works with Level II BASIC commands. Resides in only 7-Kbyte of memory May be extended indefinitely with disk-resident utilities. Supplied on 5" disk with example programs: $29.95 with instructions.

CIRCLE J Software

Two extremely useful utilities for Percom's OS-80™ DOS:

1. Machine Language Save/Load Utility. On 5" disk with bonus patch program that allows RS Renumber Utility to run under OS-80™ $14 95, with instructions.
2. VARKEEP — Adds NAME SAVE and NAME KEEP commands to OS-80™ Use one set of common data with two or more BASIC programs. Also runs under Radio Shack DOS On 5" disk, with instructions. $14.95.

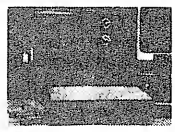
Z80ZAP

Super fast machine language disk modification utility. Read, Write, Display, and Modify sectors; remove passwords; apply patches, fixes; make backups and much more On 5" disk with instructions: $29.95.

100% machine language word processor . . . SPECIAL DELIVERY (by Software Etc.)

Use MAILFORM to create name and address lists; EXTRACT to find names by ZIP, address, gender, age, etc , SORT to sort an entire list on any field in seconds Print personalized letters written with either the Electric Pencil‡ or Scripsit* using MAILRITE Prints labels from Mailfile created under MAILFORM Runs under Percom's OS-80™, Radio Shack's TRSDOS* $125 (disk)

Inexpensive Color Graphics: the Percom Electric Crayon™

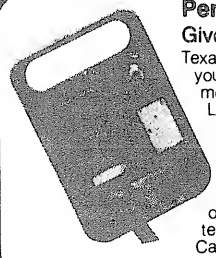


Spectacular multicolor graphics, sharp 2-color alphanumerics with your TRS-80*, a color tv and the Percom Electric Crayon™ Up to eight colors Resolution with full display memory (6 Kbytes) is 256 X 192 picture elements Microprocessor controlled, the Electric Crayon™ is not only a full color graphics system but also a complete, self-contained control computer with a dual bidirectional parallel I/O port — provision for second dual port Interface the TRS-80* via your Expansion Interface or Printer Cable Adapter. Supplied with 1 Kbyte display RAM, EGOS™ operating system and comprehensive users manual with example programs. $249 95

Optional TRS-80* interconnecting cable $24.95

Percom's Speak-2-Me-2™

Give your TRS-80* the gift of speech



Texas Instruments' Speak & Spell~ is the voice of your TRS-80* computer with this clever interface module manufactured by Percom Your own Level II BASIC programs announce, command, implore with sentences and expressions formed from Speak & Spell's~ vocabulary The Speak-2-Me-2™ PC module installs in the battery compartment of your Speak & Spell~. Power is supplied from an ordinary calculator power pak. Comes with interconnecting cable (for TRS-80* EI or Printer Cable Adapter), operating software and users manual: $69 95. (Speak & Spell ~ not included.)

Games People Play

Far out!

Scott Adams' Adventures (disks):
- Adventure Sample ..... $ 6.95
- Adventure 1-9 ..... 14.95 ea.
- Adventure 1 & 2 ..... 24.95
- Adventure 3, 4 & 5 ..... 39.95
- Adventure 6 & 7 ..... 24.95
- Adventure 6, 7 & 8 ..... 39.95

The Galactic Trilogy (disks):
- Galactic Empire $19.95
- Galactic Trader 19.95
- Galactic Revolution 39.95

Space is this hunter's domain (disks):
- Starfleet Orion $24.95
- Invasion Orion 24.95

Space Battles — from Adventure (disk):
Super fast, real-time war game with exploding missiles, full space ship control $19 95

Space Invaders — by Level IV (disk):
TRS-80* machine language version of Apple 'Super Invaders'. Sound effects $19 95

Far in!

- Datestones of Ryn (disk) $24 95
- Temple of Apshai (disk) 29 95
- Morloc's Tower (disk) 24.95

Disk System Interconnecting Cables

Improvement over RS cable design places drive 0, which includes the cable termination, at the end of the cable to eliminate the reflected noise of an unterminated cable Better data integrity Prices:
- Two-Drive Cable $24 95
- Four-Drive Cable 34.95

Power Line Filter

115/250 V, 50-400 Hz Instructions included for easy installation in standard mini-box chassis $19 95

Minidiskettes

- 10 Disks in a convenient plastic organizer box $34 90
- Single Disk 3.49

Disk Drive ID Tabs

1" x 1-¼" self-adhering plastic drive identification tabs Compatible silver with engraved black drive number Two tabs (Nos. 0, 1): $2.50; three tabs (Nos 0, 1, 2) $3 25; four tabs (0, 1, 2, 3): $4.50.

How to Order

Order by calling Access Unlimited toll-free on 1-800-527-4196†

Mail orders also accepted Orders may be charged to a VISA or Master Charge account or paid by a cashier s check, certified check or money order We accept COD orders with 25% deposit Sorry, we cannot accept personal checks We pay shipping and insurance charges on orders over $1 000 00 Add approximate insurance and shipping charges for under $1,000 00 If in doubt about these charges, ask when you call in your order Texas residents include 5% sales tax Minimum order $20 00 Allow 2 to 4 weeks for delivery

†Texas residents call (214) 494-0206

ACCESS UNLIMITED

315 N. Shiloh · Ste. D1 · Garland, TX 75042

(214) 494-0206

Prices and specifications subject to change without notice

™ trademark of Percom Data Company Inc

* RADIO SHACK and TRS-80 are trademarks of Tandy Corporation

~ trademark of Texas Instruments Corporation

‡ trademark of Michael Shrayer Software, Inc.

Circle 24

Figure 3 is a computer generated revision of Figure 2 in which our fictitious budgeteer used job 7 to request a periodic allowance of $175 and specified that account numbers 8, 9, 11, 14 and 15 be reduced accordingly. Of course if your allowance just makes these payments too small you can try again. The program has remembered your original amounts and will replace them if you say NO, the new amounts aren't satisfactory meaning that your creditors won't stand still for such small payments. You can try again, either requesting a smaller allowance or designating more accounts to share the burden of making up the deficit...or both. Perhaps, when you have been shown what the exact amounts will be, you may decide to make up the deficit out of savings...assuming you have any. In this case you will have to enter a revised amount of available resources in job 6.

There is still another option. You may choose to look the budget over and change the amounts you plan to pay certain creditors, using job 3. You then re-run the analysis (job 6) and see how big your periodic allowance will be. In Figure 4 our fictitious budgeteer has done exactly that, revising Figure 2 by changing the amounts for account numbers 9, 10, 11 14 and 15 using numbers off the top of his head. A re-run of job 6 produced Figure 4.

You don't actually write your budget from scratch each month. There are some things (like the rent or the mortgage) that you just can't change. When you get a final budget you can live with, be sure to record it (job 4). Then, next month you simply read the old one in from tape (job 5) and revise it, using job 3. You juggle the numbers until you have a spending plan that enables you $URVIVE. Hence the title.

The "Job Catalog" of $URVIVAL, (Figure 1) and the internal cues are almost completely self-explanatory, but here are some important facts about the program you must know...or at least keep handy:

NUMBER OF ACCOUNTS: Accomodates up to sixteen.

TO END YOUR INPUT OF ACCOUNTS: Respond to the cue with NG.

AUTOMATIC "YES": All questions requiring a "yes or no" answer default to "yes" when you ENTER nothing. You only respond with NO when that's your preference.

DATA RETENTION: When you run a job a second time and want the same resources or number of allowances which you had before...and this will happen plenty of times...just use ENTER without writing anything.

DELETING ACCOUNTS: (Most important of alll) Be advised this program will not, repeat, will NOT process any accounts after it encounters one with an amount equal to zero. There are two ways to remove an account from this budget:

1. Rename it "----" and give it a value of .01
2. Better yet, copy your highest numbered account over the one you want to delete. Then you can zero that account at it's old location and VOILA!, it's gone!

What about job 9? Why is that in there? Well, try answering a cue which calls for a number with an account name or some other letters....believe me....it does happen! You can get so emotionally involved in the proceedings that you get careless. You'll get a ?REDO from the computer and the screen format will be thoroughly loused up. The program still runs but things are out of kilter and hard to read. Your job catalog will be somewhere on the screen, inviting you to call job 9. Do it and everything returns to normal.

If all this scares you, copy the program and run it. True, it's a big typing job. But be assured, it's well worth the trouble. $URVIVAL is a friendly and easy program to run. The notes above will become obvious. Just don't ask for hard copy when you get the chance unless you have a printer. Otherwise you'll get hung up and lose your data.

This program runs just fine without a printer, but if you do have one, it offers you a (well protected) opportunity to request a printout of the final analysis from which you can write your monthly checks and be reminded of the periodic allowance you can draw. If you do have a printer however, there are certain things you should know. This program was written for the OKIDATA Microline-80 (see 80-U.S. for July-August 1980, page 32.) Line 1200 is calling for a 64 character print line. You will probably need to change this statement to whatever instruction your printer needs to produce a line of that length. Line 1210 is asking the printer for large characters. You have a choice of converting it to something your printer understands or omitting it altogether. You will also want to change line 1220 which returns the print size to normal (10 characters to the inch.)

A WORD ABOUT THE Z-SUBS: If you read about the Z-subroutines in the previous issue of 80-U.S. (Jan/Feb 1981), you will recognize them in the first nine steps of this program. Better yet, if you took the trouble to copy and record these useful title subroutines you can now read them into your computer from tape and begin copying this program at line 10, saving yourself at least some of the drudgery.

The program is literally peppered with GOSUB's. They are responsible for much of the neatness and smooth operation of the various jobs. The laborious way to make these things happen would be to write a much longer and more complicated program. They are as much a time saver for you in copying the program as they were for me in writing it.

If you want to explore the logic of $URVIVAL, start by examining the job catalog (which is located beginning at line 1400). Line 1520, the ON statement, contains the starting address of each job

```
BUDGET PERIOD ANALYSIS
                    JOB CATALOG
(1) INPUT BUDGET ITEMS AND AMOUNTS
(2) DISPLAY BUDGET AND TOTAL
(3) CHANGE BUDGET ITEMS OR AMOUNTS
(4) RECORD BUDGET
(5) PICK UP BUDGET FROM TAPE
(6) MAKE BUDGET PROJECTION
(7) SPECIFY WEEKLY DRAW AND ADJUST ACCOUNTS
(8) PRINTOUT BUDGET PROJECTION
(9) RESTORE HEADING
(10) EXIT PROGRAM

          WHAT NUMBER? 1
```

Figure 1

```
BUDGET FOR MARCH - 1981

1 MORTGAGE      $287.50     9 BANKCARD      $44.50
2 WATER           $8.50    10 SHELL OIL     $65.00
3 ELECTRICITY    $54.18    11 ALIMONY      $250.00
4 GAS            $37.05    12 I. MAGNIN    $171.35
5 CAR PAYMENT   $218.60    13 LODGE DUES   $125.00
6 TELEPHONE      $39.60    14 SAVINGS      $200.00
7 TAX RESERVE    $55.00    15 MOM'S GIFT    $75.00
8 GIMBELS        $38.12

SUM OF ABOVE IS $1669.40
RESOURCES ARE $2338.80
THIS LEAVES A BALANCE OF  $669.40
THIS PERMITS 5 PERIODIC ALLOWANCES OF  $133.88
```

Figure 2

```
BUDGET FOR MARCH - 1981

1 MORTGAGE      $287.50     9 BANKCARD      $29.44
2 WATER           $8.50    10 SHELL OIL     $65.00
3 ELECTRICITY    $54.18    11 ALIMONY      $165.41
4 GAS            $37.05    12 I. MAGNIN    $171.35
5 CAR PAYMENT   $218.60    13 LODGE DUES   $125.00
6 TELEPHONE      $39.60    14 SAVINGS      $132.33
7 TAX RESERVE    $55.00    15 MOM'S GIFT    $49.62
8 GIMBELS        $25.22

SUM OF ABOVE IS $1463.80
RESOURCES ARE $2338.80
THIS LEAVES A BALANCE OF  $875.00
THIS PERMITS 5 PERIODIC ALLOWANCES OF  $175.00
```

Figure 3

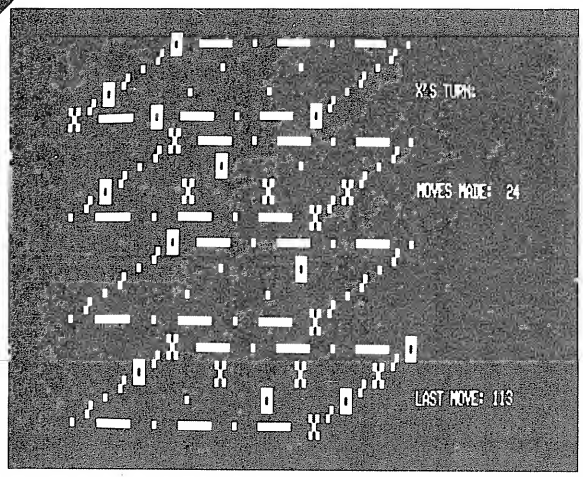
```
BUDGET FOR MARCH - 1981

1 MORTGAGE      $287.50     9 BANKCARD      $25.00
2 WATER           $8.50    10 SHELL OIL     $45.00
3 ELECTRICITY    $54.18    11 ALIMONY      $150.00
4 GAS            $37.05    12 I. MAGNIN    $171.35
5 CAR PAYMENT   $218.60    13 LODGE DUES   $125.00
6 TELEPHONE      $39.60    14 SAVINGS      $100.00
7 TAX RESERVE    $55.00    15 MOM'S GIFT    $50.00
8 GIMBELS        $38.12

SUM OF ABOVE IS $1404.90
RESOURCES ARE $2338.80
THIS LEAVES A BALANCE OF  $933.90
THIS PERMITS 5 PERIODIC ALLOWANCES OF  $186.78
```

Figure 4

(Continued on page 44)

FOR TRS-80*

QUAD



by Charles Asper

Tic-Tac-Toe, *FOUR* in a row? That's only one of the differences with Quad, Acorn's challenging three dimensional game. A graphically represented cube, four rows by four columns by four levels, Quad gives you 64 different playing positions for your X's and O's, and 76 different winning four-in-a-row combinations.

Play Quad against your computer or a friend; against the computer, there are four levels of difficulty available. You can rotate the cube six different ways to get a new perspective (or confuse your opponent!). For even more challenge, set the built-in game clock — it'll forfeit your move if time runs out!

Available for Level II, 16K. $14.95 for tape, $20.95 on disk.

These are just two of Acorn's wide selection of game, utility, educational and business programs for the TRS-80*.

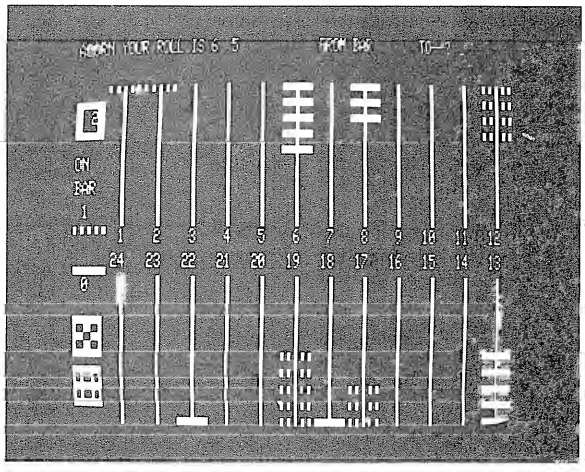
Circle 25

Acorn
Software Products, Inc.

634 North Carolina Avenue, S.E., Washington, D.C. 20003

NEW, SEMI-COMPILED VERSION!

GAMMON CHALLENGER



by Ray Daly & Tom Throop

The backgammon player featured in *Personal Computing* (August 1979) is now back in a faster, even better version! The game logic of the new Gammon Challenger has been compiled to machine language for extra speed, and there are more special features than ever.

Choose one of three levels of play, but don't get too ambitious — Gammon Challenger will put your skill to the test at all levels. For serious players, the "doubling cube" option can be used for added excitement. There *are* other computer backgammon games, but none quite like Gammon Challenger. Supplied on tape for $14.95. Requires Level II, 16K.

*TRS-80 is a trademark of Tandy Corp.

These and other popular Acorn programs are available now at fine computer stores. Ask for them.

DEALER INQUIRIES INVITED

The program was written with each job coded at the start of a hundred and contained within those hundred numbers as far as possible to make it easy to study, and possibly modify, at a later date.

The routine for printing the budget in two columns on the screen is located beginning at line 900. The tabulating variable, T, is defined at line 940 and incremented by 64 in line 960 in order to get the second column on the screen in the proper place.

Using one dimension arrays for the account names and amounts made numbering the accounts easy and helped in indentifying as well as locating data to be changed in job 3, (beginning at 300). Mathematical construction of U for use with the PRINT@ statement occurs in the last steps of the 300 series of statements and they are what cause the miraculous changes in the list to appear just where you ask for them. Personally, I hate the four second wait to record each array variable so there are some idiotic looking lines at 410 thru 430 and at 535 thru 550 to make conversions allowing the use of PRINT#-1 and INPUT#-1 which execute much faster.

Actually, $URVIVAL kills two birds with one stone. You get to learn a lot about your computer and its resident high-level language and you also get a program that will help you figure out how to meet the payments on your TRS-80....in case you are still in hock to the Bankcard or Finance company.

```
0 CLEAR 500: GOTO 10
1 Z=(64-LEN(ZT$))/2:PRINTTAB(Z)ZT$:RETURN
2 Z=960+(64-LEN(ZB$))/2:PRINT@ Z,ZB$;:GOSUB 6:RETURN
3 PRINTSTRING$(64,ZC$);:RETURN
4 PRINTSTRING$(64,CHR$(ZG));:RETURN
5 PRINT@ 64*(ZL-1),;:RETURN
6 PRINT@64*ZP,;: FOR Z=1 TO (14-ZP): PRINTCHR$(255):NEXT Z: PRI
   NT@ 64*ZP,;:RETURN
7 FOR Z=1 TO 345*ZS:NEXT Z:RETURN
8 PRINTTAB(18)"TO PROCEED TOUCH (SPACE) BAR"
9 Z$=INKEY$:IF Z$="" THEN 9ELSE RETURN
10 CLS:ZT$="BUDGET PERIOD ANALYSIS":ZG=140:ZS=2
20 ON ERROR GOTO 1340
30 R$="$$####.##"
40 DIM A$(16):DIM A(16)
50 GOSUB 1:GOSUB 4:GOTO 1400
60 PRINTTAB(45);:PRINTCHR$(27);:RETURN
100 GOSUB 6
105 PRINT "TO ENTER ACCOUNTS RESPOND TO THE PROMPTS":PRINT
110 GOSUB 8
115 GOSUB 6
120 FOR J=1 TO 16:PRINT "NAME OF ACCOUNT #";J;"= ";:INPUT A$(J)
125 IF A$(J)="NO" GOTO 1400
135 GOSUB 60:INPUT "AMOUNT";A(J)
140 GOSUB 6
145 PRINTTAB(0) A$(J);TAB(25) USINGR$;A(J)
150 NEXT J
155 PRINT"SORRY...NO MORE ACCOUNTS CAN BE ACCEPTED";:GOSUB 7
160 GOTO 1400
200 GOSUB 6:GOSUB 900
210 ZL=13:GOSUB 5:PRINT "TOTAL IS  ";USINGR$;T:GOSUB 8
220 GOTO 1400
300 GOSUB 6:IF A(1)=0 GOTO 1330ELSE 305
305 GOSUB 900
310 ZP=10:GOSUB 6:PRINT:INPUT"NUMBER OF ACCOUNT TO CHANGE";J
315 PRINT "CHANGE TITLE OF";J; "TO READ ";:INPUT A$(J)
320 PRINT "CHANGE AMOUNT FOR ";A$(J);" TO";: INPUT A(J)
325 IF J<9 GOSUB 350ELSE GOSUB 355
330 ZL=15:GOSUB 5:INPUT "ANY MORE CHANGES";Q$
335 IF Q$<>"NO" GOTO 310ELSE 340
340 Q$="":ZP=2:GOTO 1400
345 PRINT@ U,J;A$(J),USINGR$;A(J);:RETURN
350 U=128+(J-1)*64:GOTO 345
355 U=156+(J-9)*64:GOTO 345
400 GOSUB 6:PRINT"PREPARE TAPE TO RECORD DATA"
405 PRINT"RECORDING WILL START ON NEXT CUE":GOSUB 8
410 A$=A$(1):B$=A$(2):C$=A$(3):D$=A$(4):E$=A$(5):F$=A$(6)
415 G$=A$(7):H$=A$(8):I$=A$(9):J$=A$(10):K$=A$(11):L$=A$(12)
420 M$=A$(13):N$=A$(14):O$=A$(15):P$=A$(16):A=A(1):B=A(2):C=A(3
   ):D=A(4):E=A(5)
425 F=A(6):G=A(7):H=A(8):I=A(9):J=A(10):K=A(11):L=A(12)
430 M=A(13):N=A(14):O=A(15):P=A(16)
435 PRINT:PRINT"NOW RECORDING"
440 PRINT#-1,A$,B$,C$,D$,E$,F$,G$,H$
445 PRINT#-1,I$,J$,K$,L$,M$,N$,O$,P$
450 PRINT#-1,A,B,C,D,E,F,G,H
455 PRINT#-1,I,J,K,L,M,N,O,P
460 PRINT"RECORDING IS COMPLETE":GOSUB 8
465 GOTO 1400
500 GOSUB 6:PRINT"PLACE DATA TAPE IN DECK"
505 PRINT"PICKUP STARTS ON NEXT CUE":GOSUB 8
510 PRINT"DATA IS BEING PICKED UP"
515 INPUT#-1,A$,B$,C$,D$,E$,F$,G$,H$
520 INPUT#-1,I$,J$,K$,L$,M$,N$,O$,P$
525 INPUT#-1,A,B,C,D,E,F,G,H
530 INPUT#-1,I,J,K,L,M,N,O,P
535 A$(1)=A$:A$(2)=B$:A$(3)=C$:A$(4)=D$:A$(5)=E$:A$(6)=F$
540 A$(7)=G$:A$(8)=H$:A$(9)=I$:A$(10)=J$:A$(11)=K$:A$(12)=L$
545 A$(13)=M$:A$(14)=N$:A(1)=A:A(2)=B:A(3)=C:A(4)=D:A(5)=E
550 A$(15)=O$:A$(16)=P$
555 A(6)=F:A(7)=G:A(8)=H:A(9)=I:A(10)=J:A(11)=K:A(12)=L
560 A(13)=M:A(14)=N:A(15)=O:A(16)=P
570 GOTO 200
600 GOSUB 6
605 IF A(1)=0 GOTO 610ELSE 615
610 PRINT"CANNOT PROCESS UNTIL DATA IS INPUT":GOSUB 8:GOTO 1400
615 INPUT "WHAT ARE CASH RESOURCES FOR THE PERIOD";Q
620 INPUT "DIVIDE REMAINING FUNDS INTO HOW MANY PERIODIC ALLOWA
    NCES";W
```

UNIQUE !

40 & 80 Track "FLIPPY" Drives for TRS-80*

REAL VALUE

AEROCOMP offers the best value in microcomputer disc drives on the market today! Reliability, features and cost tough to beat. We deliver...and we stand behind our products, as evidenced by the only FREE TRIAL OFFER in the industry. Examine your systems needs and order today!

MYSTERY REMOVED

There appears to be some confusion in the terminology used to describe disc drives and their features. Here's what we mean:

- **FLIPPY** Allows the use of both sides of a diskette with a single-headed drive by simply turning the diskette over (model 40-1&80-1).
- **TRACK DENSITY** Specified in tracks per inch (TPI). Refers to the number of tracks per radial inch on the diskette. Typically 48 TPI=40 usable tracks and 96 TPI=80 usable tracks.
- **DOUBLE DENSITY** Refers to recording density in bits per inch (bpi). Typically single density means data can be recorded up to 2,938 bpi; double density means data can be recourded up to 5,876 bpi.
- **DOUBLE SIDED** Refers to number of read/write heads. Single-sided is one head, read/write one side only; double-sided is dual heads allowing read/write operations on both sides of the diskette. A double sided drive appears as two seperate drives to the controller.
- **ACCESS TIME** The time required for the head to move from one track to the next. Typacilly 5 to 40 milliseconds (ms).

SELECT YOURS!

- **40-Track "FLIPPY" $349.95**
(Model 40-1) 48 TPI, single dens. 125K bytes/side, double dens. 250K/side.
- **80-TRACK "FLIPPY" $459.95**
(Mod. 80-1) 96 TPI, single dens. 250K bytes/side, double dens. 500K/side.
- **40 TRACK DUAL HEAD $459.95**
(Mod. 80-2) 48 TPI, double-sided, 40 tracks/side; single dens. 250K bytes, double dens. 500K.
- **80 TRACK DUAL HEAD $599.95**
(Mod. 160-2) 96 TPI, double-sided, 80 tracks/side; single dens. 500K bytes, double dens. 1 megabytes.

NOTE: All capacity values are unformated. All models capable of single or double density. All with power supply/silver enclosure 115 VAC, 50-60 Hz 115/230 VAC available

SPECIAL PACKAGES

STARTER 40-Track FLIPPY drive,2-Drive cable,TRSDOS 2.3 Disk & Manual,Freight & Ins. $369.95 (reg. $401.00)

#1 40-Track FLIPPY drive
2-Drive cable
Newdos/80
Freight & Ins.$459.00(reg $528.00)

#2 80-Track FLIPPY drive
2-Drive cable
Newdos/80
Freight & Ins. $569.00 (reg $638.00)

#3 TWO (2) 40-Track FLIPPY drives
4-Drive cable
Newdos/80
Freight & Ins. $785.00 (REG. $893.00)

#4 TWO (2) 80-Track FLIPPY drives
4-Drive cable
Newdos/80
Freight & Ins. $999.00 (reg.$1113.85)

- **DISK OPERATING SYSTEMS**
Newdos/80 $149.00
VTOS 4.0 $125.00
DOSPLUS 3.1 $99.95

- **DISKETTES, SOFT SECTOR,5 1/4"(box of 10)**
Single-sided, single density $29.95
Double-sided, double density $39.95

- **CABLES**
2-drive $24.95
4-drive $34.95
Extender $16.95

FREE TRIAL OFFER

Order your AEROCOMP Disc Drive and use it with your system for up to 14 days. If you are not satisfied for ANY REASON (except misuse or improper handling), return it, packed in the original shipping container, for a full refund. We have complete confidence in our products and we know you will be satisfied! ORDER TODAY!

WARRANTY

We offer you a 120 day unconditional warrenty on parts and labor against any defect in materials and workmanship. In the event service, for any reason, becomes nescessary, our service department is fast, friendly and cooperative.

100% TESTED

AEROCOMP Disc Drives are completely assembled at the factory and ready to plug in when you receive them. Each drive is 100% bench tested prior to shipment. We even enclose a copy of the test checklist, signed by the test technician, with every drive. AEROCOMP MEANS RELIABILITY!

ORDER NOW!!

To order by mail, specify Model Number(s) of Drive, cable, etc (above), enclose check, money order, VISA or MASTERCHARGE card number and expiration date, or request C.O.D shipment. Texas residents add 5% sales tax. Add $5.00 per drive for shipping and handling. Please allow 2 weeks for personel checks to clear our bank. No personel checks will be accepted on C.O.D shipments-cash, money orders or certified checks only. You will receive a card showing the exact C.O.D amount before your shipment arrives. Be sure to include your name and shipping address. WE SHIP PROMPTLY! In the event there is a slight delay, you will be notified of the shipping date and we will NOT charge your bankcard until the day we ship!

WRITE AEROCOMP TODAY FOR MORE VALUES !!!

COMPARE AND BUY AEROCOMP!

	"FLIPPY"	ACCESS TIME (track to track)	HEAD LOAD SOLENOID	DISC EJECTOR	CAPACITY (unformated single density)	EASY-ENTRY DOOR	FREE TRIAL
AEROCOMP	YES	5ms.	YES	YES	250K bytes (both sides)	YES	YES
RADIO SHACK*	NO	40ms.	YES	NO	109K bytes	NO	NO
PERCOM	YES	25ms.	YES	NO	250K bytes (both sides)	YES	NO
MPI	NO	5ms	YES	YES	125K bytes	YES	NO
SHUGART	NO	40ms.	YES	NO	109K bytes	NO	NO
TANDON	NO	5ms	NO	NO	125K bytes	NO	NO

Factual materiol from current manufacturer s data sheets is believed reliable but connot be guaranteed, comparing Aerocomp Model 40-1 to similar models

The TRS-80* expansion interface limits the track to track access time to 12ms

*Trademark of Tandy/Radio Shack.

CALL TOLL FREE FOR FAST SERVICE
(800) 824-7888, OPERATOR 24
FOR VISA/MASTERCHARGE/C.O.D. ORDERS
Califarnia dial (800) 852-7777, Operatar 24. Alaska and Hawaii dial (800) 824-7919, Operatar 24.
TOLL FREE LINES WILL ACCEPT ORDERS ONLY!
For Applicatians and Technical infarmatian, call (214) 337-4346 ar drap us a card.

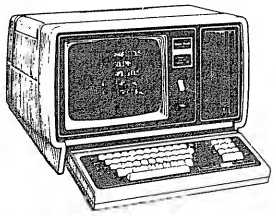
Dealers inqiries invited

AEROCOMP

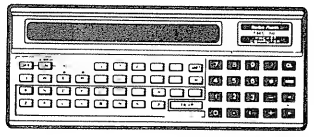
Redbird Airport, Bldg. 8
P.O. Box 24829
Dallas, TX 75224

```
625 IF Q=0 GOTO 630ELSE 640
630 GOSUB 2:PRINT "SORRY - MUST HAVE RESOURCES IN ORDER TO PROC
    ESS"
635 GOSUB 8:GOTO 600
640 IF Q<T GOTO 1310 ELSE 645
645 GOSUB 900
650 GOTO 675
655 PRINT"SUM OF ABOVE IS";:PRINTUSINGR$;T
660 PRINT"RESOURCES ARE";:PRINTUSINGR$;Q
665 PRINT"THIS PERMITS";W;"PERIODIC ALLOWANCES OF";
670 PRINTUSINGR$;(Q-T)/W:RETURN
675 ZP=11:GOSUB 6:GOSUB 655
680 INPUT "IF YOU WANT A PRINTOUT OF THIS ENTER (LIST)";Q$
685 IF Q$="LIST" GOTO 1200ELSE 690
690 GOTO 1400
700 GOSUB 6: IF A(1)<>0 GOTO 704
702 PRINT"ALLOWANCE CANNOT BE SPECIFIED BEFORE BUDGET IS ENTERE
    D":PRINT:PRINT:GOSUB 8:GOTO 1400
704 IF Q<>0 GOTO 710
706 INPUT "WHAT ARE RESOURCES FOR THE PERIOD";Q
708 PRINT:INPUT "HOW MANY PERIODIC ALLOWANCES";W
710 PRINT:INPUT "WHAT IS PERIODIC ALLOWANCE REQUESTED";V
712 GOSUB 1000
714 Y=Q-W*V-T:IF Y<0 GOTO 718
716 PRINT"THERE IS A BUDGET SURPLUS OF";USINGR$;Y:PRINT:PRINT:G
    OSUB 8:GOTO1400
718 Y=ABS(Y):PRINT"THIS WILL RESULT IN A DEFICIT OF";USINGR$;(Y
    )
720 INPUT"DO YOU WISH TO BALANCE THIS BY ADJUSTING PAYMENTS";Q$
722 IF Q$<>"NO" GOTO 726
724 Q$="":PRINT:PRINT:GOSUB 8:GOTO 1400
726 GOSUB 900:ZP=11:N=0
728 GOSUB 6
730 FOR J=1 TO 8
732 GOSUB 6:INPUT "ENTER NUMBER OF ACCOUNT TO BE ADJUSTED";S(J)
734 N=N+1
736 INPUT "ANOTHER ACCOUNT";Q$:IF Q$="NO" THEN 744
738 IF Q$<>"NO" THEN 740ELSE 744
740 NEXT J
742 DIM R(J)
744 TS=0:FOR J=1 TO N:TS=TS+A(S(J)):NEXT J
746 'SETTING ORIGINAL VALUES OF ADJUSTED ACCOUNTS ASIDE
748 FOR J=1 TO N:R(J)=A(S(J)):NEXT J
750 'PROPORTIONATE ADJUSTING OF ACCOUNTS
752 FOR J=1 TO N
754 A(S(J))=A(S(J))-A(S(J))/TS*Y
756 IF A(S(J))<0 GOTO 1350
758 NEXT J
760 Q$="":ZP=2:GOSUB 900
762 ZP=11:GOSUB 6
764 GOSUB 655
766 INPUT "WILL THIS VERSION BE SATISFACTORY";Q$
768 IF Q$="NO" THEN 770ELSE 772
770 FOR J=1 TO N:A(S(J))=R(J):NEXT J:Q$=""
772 ZP=2:GOTO 645
800 CLS:GOSUB 1:GOSUB 4:GOTO 1400
900 GOSUB 6
910 FOR J=1 TO 8:IF A(J)=0 GOTO 970
920 PRINT J;A$(J),USINGR$;A(J)
930 NEXT J
940 T=156
950 FOR J=9 TO 16:IF A(J)=0 THEN 970
960 PRINT@ T,J;A$(J),USINGR$;A(J);:T=T+64:NEXT J
970 GOSUB 1000:RETURN
1000 T=0:FOR J=1 TO 16:T=T+A(J):NEXT J:RETURN
1100 PRINT"YOU HAVE MADE AN ILLEGAL ENTRY"
1110 ZP=2:GOSUB 8:GOTO 1400
1200 LPRINTCHR$(27);CHR$(66)
1205 GOSUB 6:INPUT "FOR WHAT PERIOD IS THIS BUDGET";PP$
1210 LPRINTCHR$(31)
1215 LPRINT "BUDGET FOR ";PP$
1220 LPRINTCHR$(30):LPRINTCHR$(192):LPRINTCHR$(192)
1225 FOR J=1 TO 8
1230 IF A(J)=0 GOTO 1260ELSE 1235
1235 LPRINT J;A$(J),USINGR$;A(J);
1240 E=J+8
1245 IF A(E)=0 GOTO 1255ELSE 1250
1250 LPRINT ,E; A$(E),USINGR$;A(E)
1255 LPRINT:NEXT J
1260 LPRINTCHR$(192):LPRINTCHR$(192):LPRINTCHR$(192)
1265 LPRINT "SUM OF ABOVE IS";USING R$;T
1270 LPRINT "RESOURCES ARE";USING R$;Q
1275 LPRINT "THIS LEAVES A BALANCE OF";USING R$;Q-T
1280 LPRINT "THIS PERMITS";W;"PERIODIC ALLOWANCES OF";USINGR$;(
    Q-T)/W
1285 INPUT "ARE YOU FINISHED";Q$
1290 IF Q$="NO" GOTO 1400ELSE END
1300 END
1310 GOSUB 6:PRINT "SORRY - RESOURCES ARE INADEQUATE":GOSUB 8
1320 GOTO 600
1330 GOSUB 6:PRINT"NO DATA TO CHANGE":GOSUB 8:GOTO 1400
1340 RESUME 1100
1350 PRINT "SORRY - CHOSEN ACCOUNTS ARE NOT LARGE ENOUGH"
1360 A(S(J))=R(J)
1370 GOSUB 8:GOTO 1400
1400 ZP=2:GOSUB 6:PRINTTAB(28) "JOB CATALOG"
1410 PRINTTAB(5)"(1) INPUT BUDGET ITEMS AND AMOUNTS"
1420 PRINTTAB(5) "(2) DISPLAY BUDGET AND TOTAL
1430 PRINTTAB(5) "(3) CHANGE BUDGET ITEMS OR AMOUNTS
1440 PRINTTAB(5)"(4) RECORD BUDGET
1450 PRINTTAB(5)"(5) PICK UP BUDGET FROM TAPE
1460 PRINTTAB(5)"(6) MAKE BUDGET PROJECTION
1470 PRINTTAB(5)"(7) SPECIFY WEEKLY DRAW AND ADJUST ACCOUNTS"
1480 PRINTTAB(5)"(8) PRINTOUT BUDGET PROJECTION
1490 PRINTTAB(5)"(9) RESTORE HEADING"
1500 PRINTTAB(5)"(10) EXIT PROGRAM"
1510 PRINT:PRINTTAB(20);:INPUT "WHAT NUMBER";N
1520 ON N GOTO 100,200,300,400,500,600,700,1200,800,1300
```

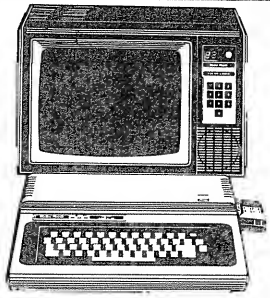
meet a REAL PLUS



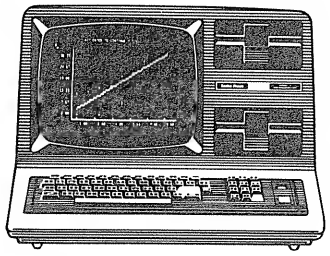
Model II 64K $3375



Pocket Computer $199



Color Computer 4K $310



Model III 16K $859

☐ **check out these unusual package values for price and warranty**

Here are just a few of our fine offers . . . call toll-free for full information.

COMPUTERS	
Model II 64K	$3375
Model III 4K LEV I	599
Model III 16K	859
Model III 32K	989
+ Model III 32K	929
Model III 48K	1099
+ Model III 48K	999
Model III 32K 2 Disc & RS232 c	2149
Color Computer 4K	310
Color Computer 16K	439.95
+ Color Computer 16K	399.95
Color Computer 16K w/extended basic	489
Pocket Computer	199
VIDEOTEX	329
APPLE 48K only	1119
ATARI 800 16K	789

PERIPHERALS	
Expansion Interface 0K	$249
Expansion Interface 16K	359.95
+ Expansion Interface 16K	339
Expansion Interface 32K	469.95
+ Expansion Interface 32K	399
16K RAM N.E.C. 200 N.S. chips	49
MODEMS	
Lynx Direct Connect	219
COMM 80 Interface	159.95
Chatterbox Interface	259.95
Telephone Interface II	169
PRINTERS	
Line Printer IV	849
Daisy Wheel II	1695
Line Printer VI	999
NEC Spinwriter 5530	2650
OKI DATA Microline 80	595
EPSON MX80	545
DISK DRIVES	
Model III 1-Drive	712
PERCOM TFD 100	389
TEAC 40 Track	319

+ Computer Plus New Equipment.
180 Day Extended Warranty

MODEL I 32K E.I., 1st. Drive, Line Printer IV, 10 Diskettes Pure R.S. $1875
combined warranty 1699

MODEL II 64K w/2 Disks, Daisy Wheel II, Scripsit Pure R.S. 6350

MODEL III 48K, 1st. Drive, Line printer IV, Scripsit Word Processor, 10 Diskettes Pure R.S. 2800
combined warranty 2699

COLOR COMPUTER 16K CC w/extended basic cassette recorder, joy sticks, dust cover Pure R.S. 569

POCKET COMPUTER w/interface Pure R.S. 239

ATARI 48K Atari 800, Atari 825 Printer, Atari 810 Disk Drive 2125

APPLE 48K Apple II & Apple II Disk w/controller, D.O.S. 3.3, Printer Card and Cable, Epson MX80 Printer 2350

PLUS real back-up warrantees —

Pure Radio Shack equipment warranteed at any Radio Shack store or dealer. Factory warrantees on Apple and Atari equipment. Other equipment carries manufacturer's warranty or Computer Plus 180 day extended warranty. Combined warrantees carry Computer Plus 180 day warranty or original manufacturer's warranty.

DEALER INQUIRIES ARE INVITED (617) 486-3193
Prices subject to change without notice.
Not responsible for typographical errors.
TRS-80 is a registered trademark of Tandy Corp.

call TOLL FREE
1-800-343-8124

computer plus

Write for your free catalog ...

Dept. A
245A Great Road
Littleton, MA 01460

Circle 26

For the Model II

Bill Schroeder

The operating system that came with the early Model II's was TRSDOS 1.1, which was just short of worthless. This was followed quickly by 1.1.2 which was just about as bad. At this point it became obvious that there were hardware problems as well as software, and modifications were provided by Radio Shack to the FDC (Floppy Disk Controller) board and the Video board.

Now we had a situation where the hardware seemed to be coming along, but the operating system was still a dead horse. This was not ignored by Tandy, and the midnight oil burned long after midnight for many nights as these problems were dealt with.

After a suitably aggravating delay, the Systems Software Group finally unleashed the not too bad TRSDOS 1.2 on us. With this release and the hardware fixes in place, everything seemed pretty rosy. Now that 1.2 has been out for over a year and has been rather well patched, debugged and otherwise healed, along comes TRSDOS 2.0! So look out, here we go again - another "new" TRSDOS to deal with.

This new operating system is Tandy's third effort at an operating system for the Model II. The amazing part of these systems is their incompatability with each other. A 1.1 or 1.1.2 cannot read a 1.2 or a 2.0, and a 1.2 cannot read a 2.0. Also when converting disks upward towards 2.0, the disk that is converted becomes readable by the new system but is no longer usable in the old system. Looking back, one would have to assume that these three generations of operating systems were each written by different companies. The formats and directories keep changing.

One manufacturer should not change an operating system to the point of incompatability with their own earlier releases. These drastic changes in an operating system would be warranted and acceptable only if they greatly increased the utility or versatility of the system. This is not the case with the evolution of TRSDOS on the Model II.

Now that the sour grapes are out of the way, let's look at what TRSDOS 2.0 has to offer, which is plenty.

The following explains the new features that have been added and the features which have been changed from the 1.2 version. I will assume a general knowledge of TRSDOS 1.2 and it's features.

SYSTEM

The following additions and changes have been made at the "System" level of the operating system:

1. ANALYZE - This command gives the user a display of the filenames on the diskette being analyzed along with the track on which the beginning of that file resides. Handy for advanced programmers, but certainly not a needed utility. This function is of little or no value to the average user of the Model II.

2. DUAL - This is a routing feature that routes all video output to the line printer as well as the video. This is a very useful feature for both the user and the programmer, and is a big plus to the system. It is unfortunate that additional routings such as (printer to file) or (video to file) or (RS 232 to printer) were not provided.

3. HELP - This command is always followed by the name of another command, such as HELP DUAL. This would display a short description of the command's syntax and all the available switches and parameters. This command is very useful to the average user and is of some value to the programmer as he becomes familiar with the system.

4. HOST - This command daisy chains the keyboard of the Model II with the RS 232 "A" receive port and also chains the RS 232 output port with the Model II's video. In effect, you would be able to run your Model II from a remote location with the use of modems and a dumb terminal. The Host utility will accept whatever SETCOM has been set on the "A" channel. This is a dumb host, as no options are provided for control of the communications or for detection of lost carrier.

5. MOVE - This powerful new command has been missing in all TRSDOS products till now. With this command it is simple to deal with moving groups of files with like

extensions or file names. The user may move all files with a /BAS extension by entering MOVE */BAS:0 TO :1. This would move all the files with the extension /BAS from drive 0 to drive 1. There are additional parameters for dealing with all the files or having the user prompted whether or not to copy each file. This command has been well thought out and is very well implemented and will become one of the most used utilities in this new system!

6. PRINT - This new command allows for the outputting of ASCII text files to the printer. Several switches are available. The LIST command with the PRT or print option supports a very similar function. The additional features in the PRINT command do not seem needed at this point as the file structure that they deal with comes from RSCOBOL and RSBASIC compiler files, but each of these provide their own printed output of their text files.

7. RECEIVE - This command allows the system to receive "object" code from another computer through the RS 232 interface. The data must be in TRS-80 load module form and then translated to INTEL paper tape standard before it is sent to the receive program. This option is handy for programmers who have not already written their own programs to deal with this, but is useless to the average user.

8. RESET - This command will cause the operating system to "Re-Boot" just as if the user had pressed the reset switch on the front panel. In effect, this is a software reset which is very handy under certain circumstances, such as ending a Basic program with SYSTEM "RESET", which will cause the computer to re-boot. This feature is of little value as several other methods of doing software resets are available.

9. SCREEN - This command will send what is displayed on the video screen to the line printer (same as JKL in NEWDOS). This command also functions fine in a Basic program line, so if the program wished to duplicate the screen to the line printer the statement SYSTEM "SCREEN" would be used. This is a very useful feature to both programmer and user.

10. - SPOOL - This is a method by which characters that would have gone to the line printer are sent to a disk file instead. At some later time, this disk file may then be spooled out to the printer as a background task, while the computer is doing something entirely different. The implementation of this spooler leaves a lot to be desired. It is not nearly as handy, powerful or dynamic as other independent spoolers, nor spoolers in other operating systems.

11. STATUS - This command displays the current top of memory as set by TRSDOS or an application, and the ON/OFF status of several of the TRSDOS functions. This command is of little value but is nice to have occasionally, especially if several different people are operating the same machine. When changing programmers I would rather re-boot before using the system.

12. T - This command performs a very nice function. It allows you to advance the line printer to the top of form, so it is ready for the next job. The printer must have been ON during boot-up or a FORMS command must have been done before this command will function.

SYSTEM CONCEPTS

TRSDOS 2.0 presents several concepts which are new to TRSDOS. First, the system is operated under what is known as interrupt mode 2, which allows the system to be interrupt driven on an "as needed" basis. The system can run somewhat faster due to the lessening of the overhead of constantly polling the I/O devices when it is unnecessary.

Second, the system now has a full type-ahead feature which makes it nearly impossible for the system to miss your keystrokes, but this feature does not function properly in Basic.

"The system is not the best operating system ever built, but is very good..."

The third, and possibly most important item, is the Alternate Directory. The system will automatically maintain a second directory on the disk which will be used should the main directory become unreadable. This feature should save much frustration.

One major change to the system involves VERIFY DETECT. This feature constantly checks that the disk in the drive has not been changed since the drive was last accessed. This is very important to users who are unable to remember to do the "I" or SYSTEM "I" commands, and ruin a disk directory as a result. The catch is that this function slows down the system terribly, but this can be solved by turning VERIFY DETECT OFF.

Another change is the implementation of "Wildcard" specfications for use with KILL, MOVE and DIR. With this feature, files can be handled by groups or classes of extension. This is one of the most important changes in the system.

The system also allows an ABS (for absolute) parameter for BACKUP, COPY, FORMAT and MOVE. This parameter tells the system not to go into the additional prompts and questions that may be associated with these commands - just to go and do it.

From the standpoint of assembler programming with TRSDOS 2.0, there have been changes to six of the supervisory calls, most of which are very minor. The big news for assembler programmers is the addition of 11 new supervisory calls. These new calls include table lookup, get directory, a sort, process the hold key, and rename a file, just to name a few. All in all, a pretty nice selection of additions for the assembler programmer.

One very aggravating thing that was done to the system was to prevent any user program from being loaded below 3000H. This means that the overlay space 2800H to 3000H is unavailable to users of assembler, unless the routines are preloaded and then block moved down to this area. This protection is useless and totally uncalled for. It just makes the system harder to work with. Protecting from 2800H on down is understandable, but not from 3000HI

CHANGES TO BASIC

A couple of changes have been made to the Basic that is provided on the 2.0 system. Two new keywords have been provided. One is ERRS$, which returns the last system number of the the last error that occured in the system along with it's error message. The other is NAME, which allows the reNAME function of the system to be used with variables from Basic.

The PRINT ZONES have been changed from 14 columns to 16. Now, when the comma is used between printed items they will be placed in 16 column boundaries.

The LPRINT function in Basic will now allow printing of up to a 255 character line. Before, only 132 character lines were permitted.

The user may now place a password on a program saved from Basic. This should have worked in the past, but did not.

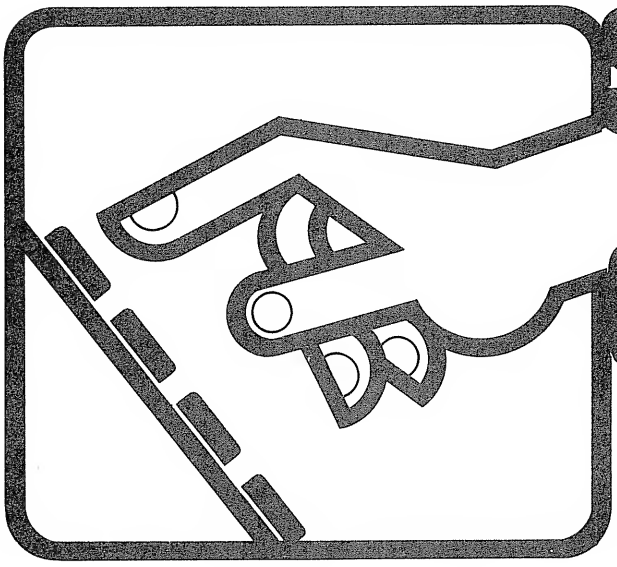
A program that contained imbeded linefeeds and has been saved in ASCII can now be loaded without encountering an error during the load.

DOCUMENTATION

The 2.0 system update comes complete with over 100 pages of new documentation. These pages are inserted per the instructions to replace a major portion of the 1.2 and old Basic manual. The documentation is excellent and is done in the same style and layout as the previous Model II documentation.

CONCLUSIONS

TRSDOS 2.0 is available from your local Radio Shack at just $24.95 including the 100 pages of documentation. I recommend every Model II user pick up a copy. The system is not the best operating system ever built, but is very good. We have used it (with slight modification) at Galactic Software as our standard "in house" system. After several months of use we are very pleased with the system's functions and reliability. Our only regret is that 2.0 was not the first operating system to come with the Model II. We can only hope that when hard disk comes along we won't have to contend with yet another incompatability problem. Until then, TRSDOS 2.0 gets my vote. ●



SYSTEM/ COMMAND

The 14th in a series.
Phil Pilgrim
Discovery Bay Software Co.
Port Townsend, Washington

When Radio Shack announced their new Level II ROM for the Model I, I immediately wondered, "Oh no, what did they change?" Like many machine language programmers, I use some of the unpublished routines in ROM to simplify programming, not only for myself, but also for this column and the programs I sell. A change in the ROM might mean having to rewrite many routines to accommodate those users with the new ROM. Radio Shack's insistence in their newsletter that they bloody well had the right to scramble up any unpublished addresses they pleased only added to my anxiety.

Recently I had the opportunity to examine the new ROM and to compare with the old one. I'm happy to say my worst fears have not materialized. In a departure from the usual project orientation of this column, I want to share with you my discoveries, not only to point out the differences, but to reassure you that any addresses *not* mentioned here have not been changed.

First of all, how do you know which set of ROM you have? If LEVEL II BASIC comes up "R/S L2 BASIC", you have the new set; "RADIO SHACK LEVEL II BASIC", the old one. Functionally, the major changes have been to add keyboard debouncing, improved cassette reading, and the ability to use @ anywhere in a PRINT statement. The result *could* have been to shift most of the addresses in ROM one way or the other to accommodate the changes. Thankfully, the modifications are largely self-contained. In the following paragraphs, I will show which bytes have been changed, and try to explain what the change accomplishes, though in some cases, I haven't the foggiest idea.

0059

Changed from 1A to 00. The effect of this change is to inhibit a CHR$(26) from being returned from the keyboard routine when you type a *SHIFT-down arrow*. This allows these two keys to be used as a control key. Any alphabetic key struck while these two keys are held down will yield the control equivalent of that letter. CNTL-H, for example, returns a CHR$(8) or Backspace.

00FD

Changed from 11 to 0E. This is just part of a LD HL instruction which gets the address of the Level II Basic sign-on message for display on the screen. That message now starts at 010E instead of 0111.

0108 - 021C

Changed from "ORY SIZE RADIO SHACK LEVEL II BASIC (Carriage Return)" to "SIZE R/S L2 BASIC (Carriage Return)". The space freed up between 011C and 012C is now occupied by the new debounce routine.

0249 Changed from 41 to 60.
0250 Changed from 76 to 85.

Both of these bytes deal with timing during a tape read. The first change lengthens the delay after the timing pulse is detected before resetting the cassette input data latch. The second, lengthens the wait for the data pulse. This apparently makes marginal cassettes a little easier to read.

03FB - 03FD

This is part of the keyboard routine. The code originally here has been replaced with a JP to the debouncing patch at 011C. This patch returns with a JP to 03FE.

124C - 124D

The two one-byte instructions at these locations have been interchanged. I can't find any references to 124D, so I can only assume the effect is nil.

206D - 20F7

This is the portion of the PRINT routine which handles

the @ and #, among other things. The effect of the change is to allow the @ anywhere in the PRINT, rather than just at the beginning. The logic which accomplishes this can be summaraized as follows:

Old Version

2072: Is next character an @?
No: Go to 208F
2076: Yes: Position cursor on screen.
208F: Is next character a #?
No: Go to 209B.
2093: Yes: Turn on cassette motor; write leader.
Direct output to cassette.
209B: Do other PRINT stuff.
20F6: Loop back to 209B.

New Version

2072: Is next character a #?
No: Go to 207C.
2076: Yes: Turn on cassette motor; write leader.
Direct output to cassette.
207C: Do other PRINT stuff, including the @ routine from 2084 to 20A3.
20F6: Loop back to 207C.

213B

Changed from 3F to 7F. The byte at this address is ANDed with the TAB parameter in a PRINT or LPRINT to get the new print or display position. Contrary to the Level II Manual, the old version allowed tabbing only within the current display line (up to position 63). The change allows tabbing up to position 127 (position 63 on the *following* display line). This should be a boon to those with wide printers.

2167

Changed from A0 to 81. This is part of a JP instruction. The old jump address was 20A0. The new one is 2081. Codewise, these are equivalent locations between the two ROMs.

226A - 226E

A check of location 40A9 has been replaced with NOPs. What this check originally accomplished and why it was eliminated, I have no idea.

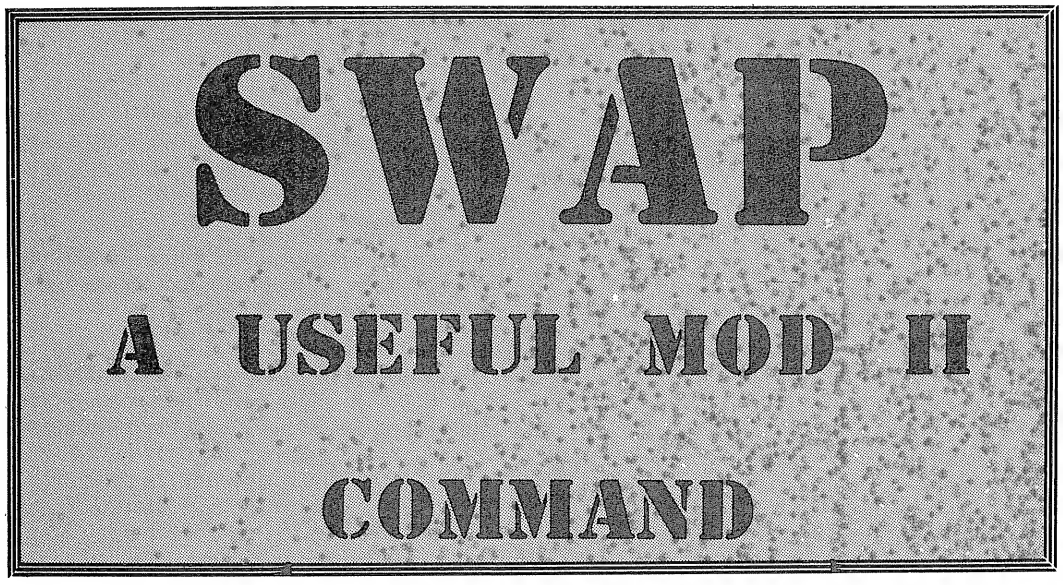
2C1F - 2C42

This is the beginning of the CLOAD and CLOAD? routine. The code at these addresses has been modified from starting up the cassette motor and finding the sync byte and *then* getting the parameters of the CLOAD and performing a NEW, to getting the parameters and performing the NEW first, *then* turning on the cassette, etc. The only reason I can find for doing this is one of timing. If the parameters took too long to determine, some data might be missed with the motor running.

2FFC - 2FFF

Changed from zeroes to apparent garbage. These are most likely just leftover bytes at the end of ROM.

So there you have it. If you use code at any of the addresses mentioned, beware of the changes if you intend your programming to be transportable. Otherwise, breathe easy. Things ain't so bad after all. ●



T R Dettmann

Many of the current users of the Model II came from the Model I originally. As such, they uploaded many programs from that system to use on their new systems. This affected programmers as well as users, since many of the programs written for business applications on the Model I were uploaded and sold with only necessary modifications for the Model II.

One area where this is very apparent is in sorting applications. On the Model I, it was normal to use the program statements:

T$=A$(I):A$(I)=A$(J):A$(J)=T$

to sort an array into order. This causes a problem with Microsoft Basic, since each time the statement is executed, three new strings are created in string memory. Soon, string space is full and there are many unnecessary strings that have to be eliminated because they are duplicates.

In this case, Microsoft Basic goes into a memory management mode and cleans up the string space, compacting into the smallest available space and freeing up for more strings. In a sort, this appears as a hesitation, sometimes almost like a lock-up, depending on how much space there is to clean up.

To correct this on the Model I, we suggested (*80-U.S. Journal, Nov/Dec 79*) that you use a VARPTR technique to prevent having to create new strings. This technique simply reassigned the names to the existing strings using PEEK's and POKE's. That was okay on Model I, but Model II dosen't have PEEK or POKE in its Basic.

Model II Basic provides something much cleaner and easier to use than the old reassignment technique above. It is called "SWAP".

Using SWAP, two variables can be reassigned names quickly without having to create new strings. This will save time in a sorting program. This will not make an inefficient sorting technique any better, but it will make it the fastest it can be on the Model II without going to assembly language.

The listing with this article is a program used to test the SWAP command on the Model II. Line 70 sets aside the size of memory reserved for strings. By changing the size of the string memory, we were able to vary the time needed to go through the first test loop, but the time through the second loop was unaffected.

The results of testing are summarized in the following table:

TABLE 1

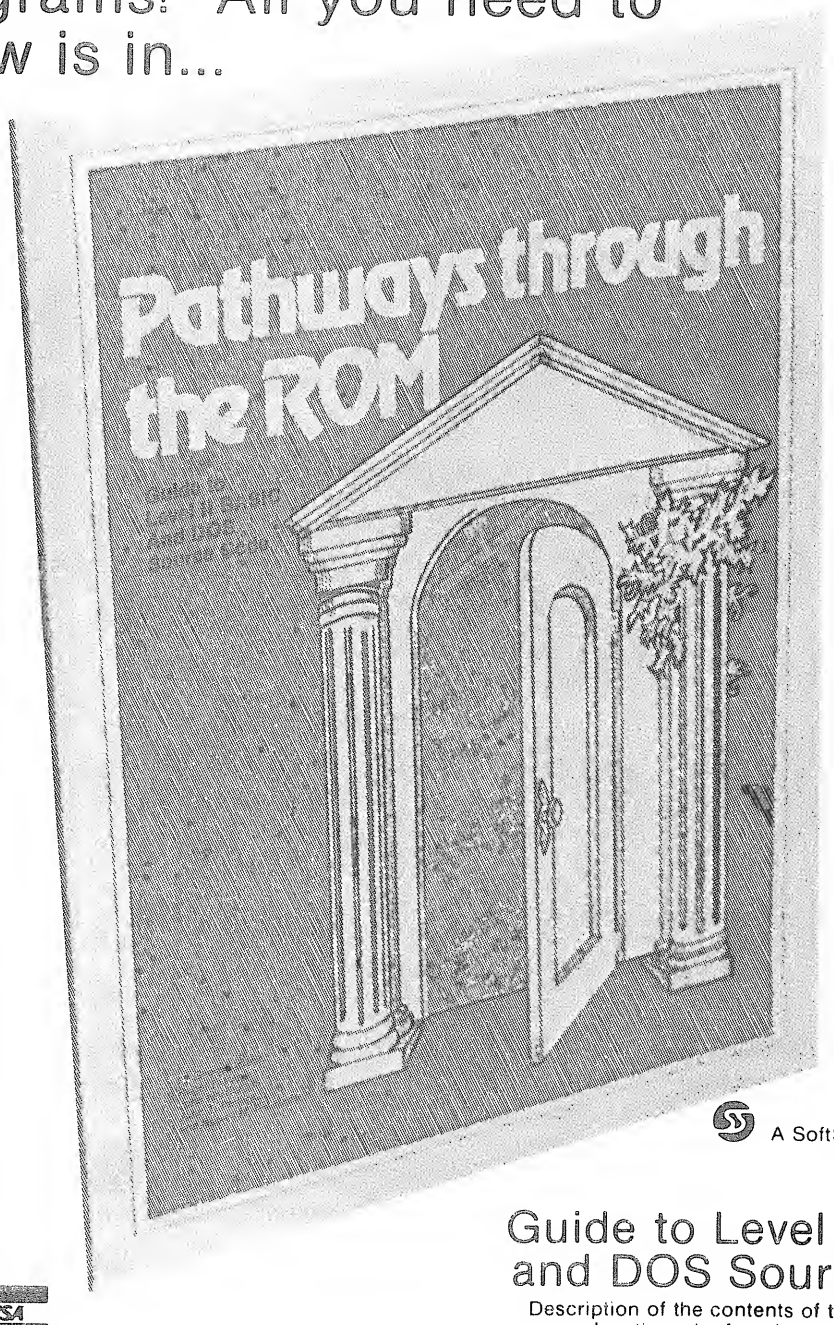
CLEAR	Test 1 Time	Test 2 Time
200	26 sec.	3 sec.
500	11	2
5000	8	3
10000	8	3
30000	8	3

If you throw in more complicated string reassignments such as you have during a sort, these numbers will get worse, and the difference between SWAP and the brute force method will get larger.

Running the same test on non-string variables shows a slight (2:1) advantage for SWAP over the brute force method. This difference didn't change depending on CLEAR size or any other variation in the program. The SWAP took 3 seconds for 1000 iterations and the brute force method took 6 seconds. ●

```
10 REM*************************************************
20 REM
30 REM  TEST OF SWAP COMMAND
40 REM  BY TERRY R. DETTMANN
50 REM
60 REM*************************************************
70 CLEAR10000:REM                    SET THIS TO WHAT YOU WANT
75 REM              THE DUMMY STRINGS
80 A$=STRING$(50,"A"):B$=STRING$(50,"B")
90 PRINT"START OF TEST 1 AT: ";TIME$
95 REM              LOOP WITH THE BRUTE FORCE SWAP APPROACH
100 FORI=1TO1000
110     T$=A$:A$=B$:B$=T$
120 NEXTI
130 PRINT"END OF TEST 1 AT: ";TIME$
140 PRINT"START OF TEST 2 AT: ";TIME$
145 REM              LOOP WITH THE SWAP INSTRUCTION
150 FORI=1TO1000
160     SWAP A$,B$
170 NEXTI
180 PRINT"END OF TEST 2 AT: "TIME$
190 END
```

Unlock the hidden power of your computer for fast and easy programming! Use ROM routines in your BASIC and Assembly Language programs! All you need to know is in...



INCLUDES:

SUPERMAP
From Fuller Software ($18.95)

TRS-80 DISASSEMBLED HANDBOOK
by Robert Richardson ($10.00)

HEX MEM
by John Phillipp
Monitor written in BASIC

Z-80 DISASSEMBLER
by George Blank

ALL ONLY $19.95
plus $1 shipping

A SoftSide Publication

Guide to Level II BASIC and DOS Source Code

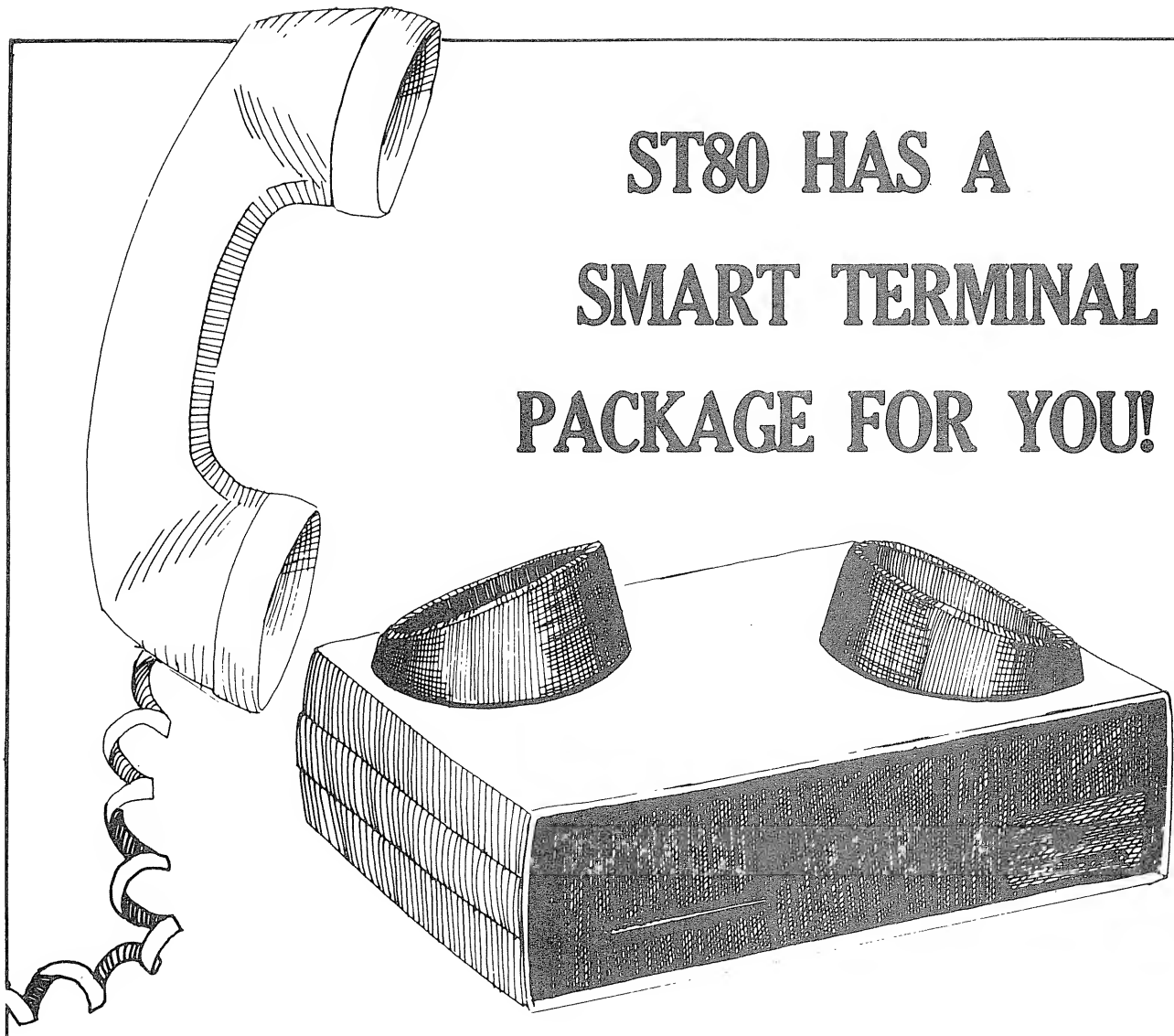
Description of the contents of the Level II BASIC ROM by memory locations, by function, and in lesson format. Includes several BASIC and Assembly Language programs in listing format to examine and use ROM routines.

MasterCard VISA

ORDER TOLL-FREE
(In NH call 673-5144)
1-800-258-1790

The Software Exchange

Circle 28



ST80 HAS A SMART TERMINAL PACKAGE FOR YOU!

All four programs include the ability to use an unmodified TRS-80™ keyboard to produce RUB, ESC, and other control characters for time sharing, software control of the RS-232-C board, repeat key, bell, software support for the three most common upper/lower case hardware conversion, and line printer output.

ST80* UC

Preset parity, word length, and baud rate (regardless of switch settings on the RS-232-C board) for THE SOURCE, MICRONET, and FORUM 80, automatic testing of the RS-232-C board, and even spooling of prepared messages on tape directly into FORUM 80 using a basic program supplied as a line listing. 4K Level II cassette, ... $24.95

ST80*

Reprogram your RS-232-C board from the keyboard, and run at different baud rates. Note: does not have auto testing of the RS-232-C or tape spooling. 4K Level II cassette $49.95

ST80* D

Connection time clock, option of user-created translation tables for keyboard, gathering and pre-formatting data to be sent directly from disk to host computer, spooling of received files to disk or printer, editing of received files, and auto logon. Use it with VTOS 3.1, and get device-driven I/O, job logging, and chaining. 32K disk program, ... $79.95

ST80* III

ST-80 D with extra utility programs. 32K disk program, ... $150.00

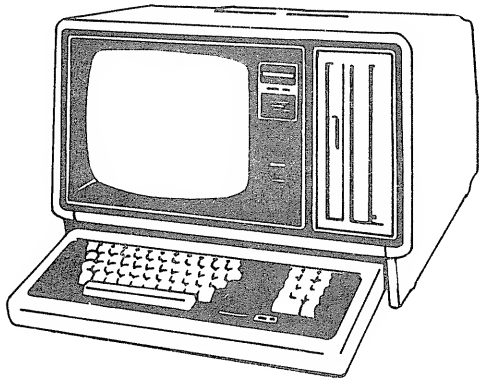
TSE-HARDSIDE

6 South St., Milford, NH 03055 (603) 673-5144

TOLL FREE OUT-OF-STATE 1-800-258-1790

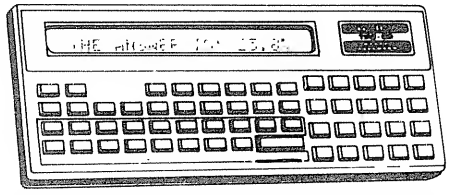
* A trademark of Lance Micklus, Inc.

Circle 29



MODEL II $3599.00

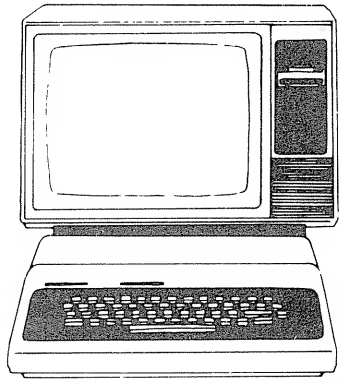
	LIST PRICE	OUR PRICE
Model II, 64K RAM	$3899.00	**$3599.00**
Model III, 16K RAM	$999.00	$929.00
Model III, 32K Dual Disk	$2495.00	$2299.00
Pocket Computer w/Interface	$298.95	$269.00
TRS-80 Color Computer	$399.00	$359.00
TRS-80 Color Computer Expanded	$599.00	$519.00
COMM-80 Interface	$179.95	$159.95
CHATTER BOX Interface	$279.95	$259.95
DISK-80 Interface	$349.95	**$339.95**
Expansion Interface, no RAM	$299.00	$279.00
Expansion Interface, 16K RAM	$418.00	$339.00
Expansion Interface, 32K RAM	$537.00	$399.00
RS-232-C Board	$99.00	$89.00



POCKET COMPUTER $269

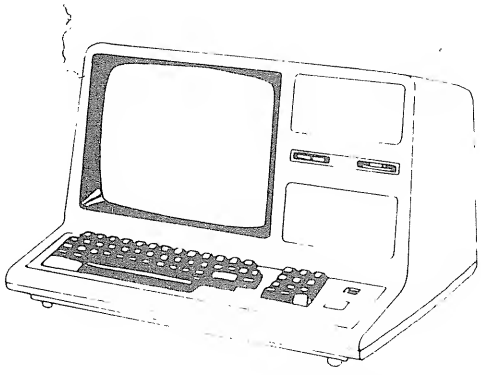
with interface

	LIST PRICE	OUR PRICE
TRS-232 Printer Interface		$59.95
16K Memory Kit, TRS-Keypad	$119.00	$59.00
16K Memory Kit, TRS-Exp. Int.	$119.00	$59.00
Upper/lower Mod Kit	$59.00	**$25.00**
Video Reverse Kit		$23.95
CPU Speed-up Kit		$24.95
Percom Electric Crayon, w/cable		$279.95
TRS-80 Dust Cover (3pc set)	$9.95	$7.95
TRS-80 Computer Case	$109.00	$99.95
TRS-80 Monitor Case	$84.00	$84.00



COLOR COMPUTER $359

	LIST PRICE	OUR PRICE
Percom, TFD-100, 40-track	$429.95	$399.00
Percom, Dual TFD-100 Drives	$849.00	$799.00
Percom, TFD-40, 40-track	$399.95	$379.00
Percom, TFD-200, 77-track	$675.00	$629.00
Hardside 40-Track Disk Drive	$399.00	$329.00
Hardside 80-Track Disk Drive	**$499.00**	**$449.00**
Percom Data Separator		$29.95
Extender Card	$15.95	$15.00
2-Drive Cable	$29.95	$29.00
4-Drive Cable	$39.95	$39.00



MODEL III $929

TERMS:

TERMS: Prices and specifications are subject to change. HARDSIDE accepts VISA & MASTERCARD. Certified checks and Money Orders; Personal checks accepted (takes 3 weeks to clear). HARDSIDE pays all shipping charges (within the 48 states) on all PREPAID orders OVER $100.00. On all orders under $100 a $2.50 handling charge must be added. COD orders accepted (orders over $250 require 25% deposit) there is a $5.00 handling charge. UPS Blue Label, and Air Freight available at extra cost.

TO ORDER TOLL-FREE:

1-800-258-1790

(In NH call 673-5144)

MasterCard VISA

HARDSIDE™

Circle 30

APL

by Phelps Gates

Now a high-level, scientific programming language for the home computer that doesn't cost $200 or $300. The power of this language is in its strong mathematical operations, especially with regard to matrices and vectors. Programs requiring matrix multiplication or other matrix problem solving that would require hours of programming time in BASIC are solved quickly and with minimal effort in APL.

To aid in learning APL, lessons are included on the disk. Starting from the basics, you are brought step by step through the various programming techniques involved with APL. These lessons act as a tutor which will have you "talking APL" in no time. Also available is the book, "APL: An Interactive Approach," which reinforces many of the examples given in the lessons and provides additional insight into APL programming.

FEATURES

APL-80 on disk contains the following features:)SAVE and)LOAD workspace on disk;)COPY other workspaces into current ones; Return to DOS for directory or commands without losing your workspace; Send output to lineprinter; Five workspaces of lessons included; Sequential and random files; 15 digit precision; Monadic and dyadic transposition; Easy editing within FUNCTION lines; Latent expression (FUNCTION can "come up running" when loaded); Tracing of function execution; Real-time clock; User-control of random link; Workspace is 25587 bytes (in 48K machine); Arrays may have up to 63 dimensions.

COMMANDS APL-80

APL-80 supports the following commands; Absolute value, add, and, assign, branch, catenate, ceiling, chr$/asc, circular, combinatorial, comment, compress, deal, decode, divide, drop, encode, equal, expand, exponential, factorial, floor, format, grade down, grade up, greater, greater/equal, index generator, indexing, index of, inner product, label, less, less/equal, logarithm, maximum, member, minimum, multiply, nand, negate, nor, not, not equal, or, outer product, peek, poke, quad, quote quad, random, ravel, reciprocal, reduction, reshape, residue, reverse, rotate, scan, shape, sign, system, subtract, take, transposition.

SPECIFICATIONS

Minimum system requirements: 32K disk system (48K recommended) includes APL-80, Five workshapes of lessons, instruction manual.
Price ..... $39.95 on disk
Reduced feature: 16K Level II tape version, no lessons.
Transpositions, format, and inner product not implemented. Reduced domain for some functions, 6 digit accuracy.
Price ..... $14.95 on cassette
APL: An Interactive Approach
Price ..... $16.95 ($3.00 shipping charge)

LIMITATIONS

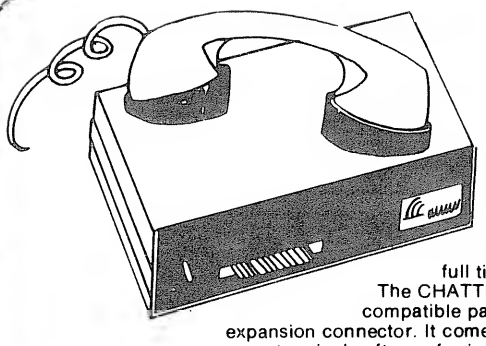
Due to the absence of the special APL character set on the TRS-80™, APL-80 uses shifted letters to represent the various APL characters.
In addition to the keyboard limitations, lamination, domino, and matrix inverse are not implemented but can be derived with user-defined functions.
Multiple specifications must be split into two statements unless the left-hand assignment is to a quad. This also applies to implied multiple specifications.
Reduction and reshape (p) are not permitted for empty arguments; the argument of add/drop may not be scalar; empty indices are not permitted.
A quad (q) can't be typed in response to a quad (nor can the name of a function which itself gets input from a quad). Quote-quad (m) is permitted.
No more than 32 user functions can be defined in a single workspace and a function may not contain more than 255 lines.
A comment (c) must occupy a separate line: a comment can't follow a function statement on the same line.
In the tape version, arrays are limited to five (5) dimensions.

TSE-HARDSIDE

MasterCard VISA

6 South St., Milford, N.H. 03055
ORDER TOLL FREE: 1-800-258-1790
(in NH call 673-5144)

Circle 31



The Chatterbox

A TRS-80 Interfacing Alternative

The CHATTERBOX is a unique packaging combination of the presently available COMM-80 I/O Interface for the TRS-80* and an acoustic modem. This one box is all that is required to turn even a barebones 4K TRS-80* into a full time–sharing terminal.

The CHATTERBOX includes built-in programmable 50-19200 baud serial port, a Centronics compatible parallel printer port, a 300 baud acoustic originate modem, and a spare TRS-BUS expansion connector. It comes complete with power supply, ribbon cable and connector, user's manual, and terminal software for immediate operation. When the modem is in use, the complete data conversion is automatically routed to the serial output port where it can be logged on a printer.

The CHATTERBOX is the only peripheral needed to allow a TRS-80* to communicate with time–sharing systems such as MICRONET and the SOURCE.

It is completely hardware and software compatible with existing TRS-80* products and connects either to the keyboard connector or screen printer port on the RS Expansion Interface. Features: Full 8-bit parallel port; RS-232-C serial port (up to 19,200 baud); Acoustic modem; TRS-BUS connector for future expansion; Connects to Keyboard or E.I.; Includes terminal software; Users manual; Power supply. $259.95

PRINTERS

	LIST PRICE	OUR PRICE
Centronics 730	$795.00	$749.00
Centronics 730-3	$895.00	795.00
Centronics 737	$995.00	$869.00
Centronics 779	$1395.00	$1095.00
Centronics 779 w/lower case	$1595.00	$1195.00
NEC 5510 SpinWriter		$2795.00
NEC 5520 SpinWriter		$3195.00
NEC 5530 SpinWriter		$2695.00
NEC Tractor-Feed Option		$250.00
LRC 7000® (64-col.)	$405.00	$299.00
Okidata Microline-80	$800.00	$639.00
Tractor-Feed Option	$140.00	$129.00

LRC to TRS-80	$20.00
LRC to PET, IEEE	$59.00
LRC to RS232C, male or female	$65.00
730 or 737 to TRS-80	$29.00
NEC or 779 to TRS-80	$35.00
RS-232-C to RS-232-C, male to male	$24.95

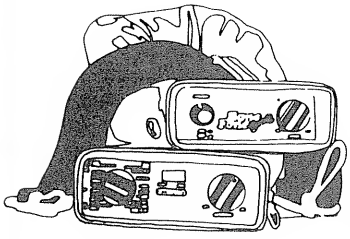
GENERAL INTEREST

	LIST PRICE	OUR PRICE
BSR System X-10 Starter kit	$124.95	$99.95
BUSY BOX, TRS-80	$114.95	$99.95
BUSY BOX, S-100	$119.95	$114.95
AC-SFK-31 Line Filter	$24.95	$19.95
ISO-2 Line Filter & Isolator	$56.95	$49.95
ISO-2/CBS Line Filter-Isolator	$70.95	$59.95
ISO-7/CB Super Filter-Isolator	$146.95	$99.95
Mini-Flex Diskette File	$24.95	$19.95
CASIO C-80 Calculator Watch	$49.95	$44.95
BONE FONE	$69.95	$56.95
LOGOS-9 Printing Calculator	$99.95	$79.95

SPECIAL

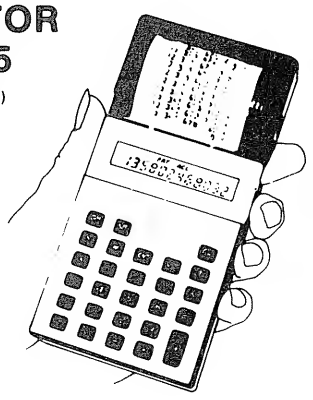
CASIO C-80 CALCULATOR WATCH $44.95

(plus $2.50 postage and handling)



BONE FONE $56.95

(plus $2.50 postage and handling)



OLIVETTI LOGOS-9 PRINTING CALCULATOR $79.95

(plus $2.50 postage and handling)

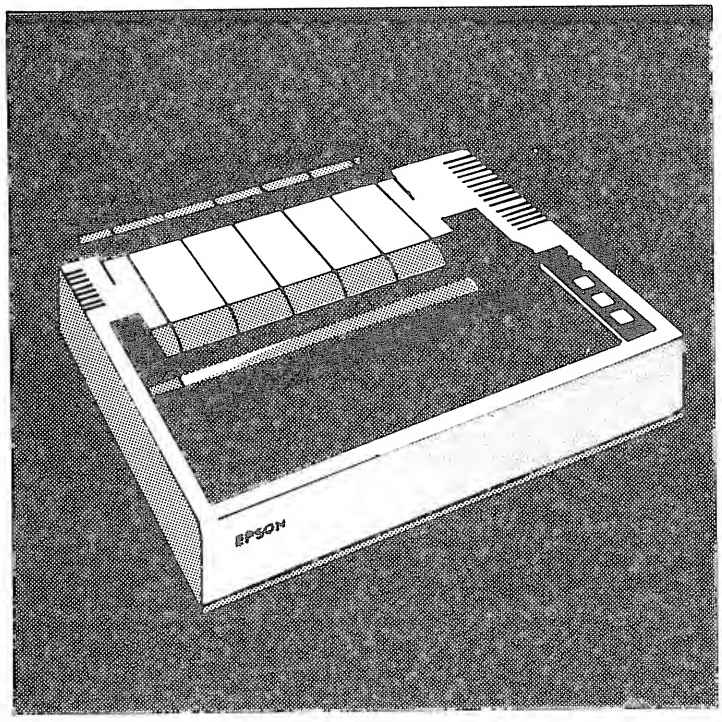
HARDSIDE™

TO ORDER TOLL-FREE
1-800-258-1790
(In NH call 673-5144)

MasterCard VISA

Circle 32

If you just bought another printer, boy are you gonna be sorry.



MX-80

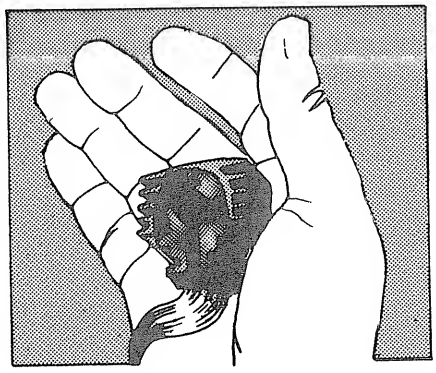
The Epson MX-80. It's not just another worked-over rehash of last year's model. It's our top-of-the-line 80-column printer. It's new. From the ground up. And it's the most revolutionary printer to hit the market since Epson invented small printers for the 1964 Olympics in Tokyo. Don't take our word for it, though. Compare. There simply isn't a better value in an 80-column printer. Period.

But here's the fact that's going to stand the printer world on its ear. The MX-80 sports the world's first disposable print head. After it's printed about 50 million characters, you can throw it away. Because a new one costs less than $30, and the only tool you need to change it is attached to the end of your arm.

Now that's revolutionary, but that's only the beginning. The MX-80 also prints bidirectionally at 80 CPS with a logical seeking function to minimize print head travel time and maximize throughput. It prints 96 ASCII, 64 graphic and eight international characters in a tack-sharp 9x9 matrix. And it provides a user-defined choice of 40, 80, 66 or 132 columns and multiple type fonts.

We spent three long years developing the MX-80 as the first of a revolutionary series of Epson MX Printers. We employed the most advanced automatic assembly and machining techniques in existence to produce a printer that is incredibly versatile, remarkably reliable and extraordinarily inexpensive. It's a printer that could only come from the world's largest manufacturer of print mechanisms: Epson.

If it sounds like we're proud of the MX-80, we are. Not only does it do things some of the world's most expensive printers can't do, it'll do them for you for $569. That's right ..... $569.



The world's first disposable print head. It has a life expectancy of over 50 million characters, yet it's so simple, you can change it with one hand. And it cost less than - repeat less than -$30.

TSE·HARDSIDE

6 South St, Milford, NH 03055 (603) 673-5144

TOLL FREE OUT-OF-STATE 1-800-258-1790

Circle 33

Gain Copy Flexibility with Newdos 80

Available for Model I from Apparat Inc.

David Busch
Revenna, Ohio

So near, and yet so far. Your friend has finished writing a machine-language utility program for your TRS-80 Model 1 that will save a great deal of time in a project you plan to work on tomorrow. Luckily, he remembered to bring the right diskette when he came to your home to visit. You load the disk, type in the file name, and wait. Your single disk drive sputters for a few seconds, then flashes a helpful NO SYSTEM message on the CRT screen.

"Oh, I forgot -- that's a data diskette, and you only have one drive," your friend says, sheepishly. "I guess there's no way you can get at it. Wait! If we loaded Superzap..."

You shake your head and smile. You boot up again after inserting your NEWDOS 80 disk, and type:

COPY :0 $UTILITY/CMD TO UTILITY/CMD

Two disk swaps later, your friend's utility is safely deposited on your own system disk, ready for use. The friend nods his head in appreciation. You don't need more than one drive to access data diskettes.

For those of us who grew up on TRSDOS 2.1 and still remember having to carefully bulk erase disks in order to re-use them, the COPY utility built into NEWDOS 80 is nothing short of amazing. Just the ability to COPY individual files located on disks with no system at all can be a lifesaver -- for example when a two-drive owner has to take one drive in for repair.

There are six formats to NEWDOS 80's COPY command. Let's look at each, and some (only *some*) of the things you can do with them. The documentation included with NEWDOS 80, while clear and complete, tends to understate some of the interesting possibilities.

Format 1. Copy a file from one place to another. The name of the file can be changed (TEST to TEST/BAS, for example), or left exactly the same. The new file can be created on the same disk as the old (as long as a new name is specified), or on another disk in a different drive.

The interesting part of all the NEWDOS 80 COPY formats is that much of the input is optional. One can enter:

COPY,TEST:O,TO,TEST/BAS:1

or

COPY TEST:O TEST/BAS:1

Each accomplishes exactly the same thing. In this case, a space was used instead of the commas, and the optional TO left out. If the file name is to be the same in both files, then the command can be even shorter:

COPY TEST/BAS:O :1

This format is a good, basic copy utility. It provides no prompts to mount source or destination disks in their proper locations. The user must already have the disks in the right drives.

Format 2. This is the format that allows a two-drive owner to copy files from data disks onto other data disks, or onto system disks. A disk with no system, or a "foreign" system can be mounted on Drive 0 with no problems. The "$" prefix on the file specification performs the magic. To be entirely accurate, the $ informs the system that either the source and/or destination disk which will be mounted in Drive 0 will either have no system or one that is not identical to the one most recently booted up. See Format 4 for a similar implementation for one-drive owners.

Format 2, like 3, 4, 5, and 6 provides helpful prompts to let the user know which disks are to be mounted where and when. Other than the dollar sign, the syntax is identical to that of Format 1.

Format 3. This is the standard copy format for making a copy, either from one disk to another, or from one file name to another file name on the same disk, using just one drive. The basic syntax is this:

COPY,:O,TEST,TO,TEST/BAS

or

COPY,:1,TEST,TO,TEST/BAS

In either case, Drive 0 must contain a system disk identical to the one most recently booted up. The source and destination drive must be the same, and, therefore, can't be specified in the file name. This format is suitable for copying files from one system disk to another if you have only one drive. Those with two drives will find this useful for copying files from system or data disks to another disk on Drive 1 leaving the system disk in Drive 0 at all times. As always, much of the input is optional:

COPY 1 TEST TEST

This will copy a file named TEST in Drive 1 to a second destination disk inserted in Drive 1 at the proper time. Note that the colon on the drive spec can be left off.

Format 4. This format permits one-drive owners to access programs on data disks (if they know the file name), and transfer them to system disks or other data diskettes for use. The command looks like this:

```
COPY,:O,$TEST/BAS,TO,TEST/BAS
or
COPY O $TEST/BAS TEST/BAS
```

Apparat's documentation claims that Formats 2 and 4 make it practical for suppliers of programs to send out their wares on data disks to customers who own only one drive, thereby avoiding the illegal act of providing those purchasers with free Disk Operating Systems on the program disks they buy.

This only works, of course, if the end user owns NEWDOS 80 already. Seems like circular logic to me.

Format 5. This is where NEWDOS 80's COPY utility starts to get interesting. This format allows making a full diskette copy, with many options. I'll try to explain a few of them. The command can be as simple, or as complex as the following:

```
COPY 0 1 01/02/81
or
COPY,:0=35,TO,:1,01/02/81,NDMW,NFMT,
     PASSWORD ,NPASSWORD,KDN,KDD,
```

Parsing the last example shows what will happen. The source disk, a system disk with NEWDOS 80, is already mounted on Drive 0, and the destination disk is mounted on Drive 1. There will be no prompts to mount the disks, because NDMW has been specified (no diskette mount waits).

The user wishes to copy only 35 tracks of the source disk. Most likely, this would be because the source disk had only 35 tracks.

Under NEWDOS 80, each drive's default track count can be set using the PDRIVE command. Then, unless the user tells the system differently, DOS will always assume that the default figure should be used for copying, formating, etc. The ":0=35" option allows changing this.

In this example, assume that both Drive 0 and Drive 1 had been defined as 40-track drives. The system will only copy 35 tracks from Drive 0 onto Drive 1. Unless NMFT (do not format the disk) is specified, NEWDOS 80, after copying the specified number of tracks, will format the remaining tracks up to the default number of the destination drive. This step can be avoided by defining the tracks of the second drive in the COPY command (i.e. :1=35).

NMFT speeds up full disk duplication when copying to disks that have previously been formatted.

The next two items of input show PASSWORD and NPASSWORD. This assumes that passwords have been enabled (via the SYSTEM command) and are required for copying. (Under NEWDOS 80, you have the option of requiring passwords for accessing a file, while permitting protected files to be copied without a password).

In this example, PASSWORD is the current password of the source disk, while NPASSWORD is the new password that will be placed on the destination disk. If no new password is specified, the same password as used on the source disk will be applied.

KDN and KDD tell the system to allow the destination disk to keep its old name and date, rather than acquire those of the source disk.

The following are some of the options not used in the example. Many had to be left out because they are mutually exclusive with one of the other options.

BDU will be especially useful for those learning to use SUPERZAP. With this choice in operation, the system skips the normal final step of updating the disk directory with name, date, password, and other information. Apparat says that BDU is useful when the source diskette "has a bad directory, a non-standard directory, or has no directory at all." What it does is give you the chance to make several good copies of disks with bad directories. Then you can apply SUPERZAP to one of the copies, attempting to repair directory entries and fooling around with HIT and GAT sectors. For obvious reasons, BDU cannot be used with KDN, (keep disk name), NDPW (new disk password), NDNK (new disk name), or USD (use system date.)

Other options which cannot be used with NDMW allow the user to specify certain conditions on the destination disk before the copy will proceed. Disks with data or with the wrong name can stop the copy while the operator decides whether to proceed.

UBB (use big buffer) will be helpful for owners of 8-inch disk drives. As you will have noticed, two-drive copies proceed differently than their one-drive counterparts. When making a two-drive copy with 5¼" diskettes, the system duplicates one track at a time, loading 10 sectors into memory, writing it to the destination disk, verifying, and then loading the next track.

This wouldn't work with a one-drive copy -- such a system would require loading the source disk once for each track to be copied. Instead, all available memory is used to load many tracks - typically 15 or so with a 48K system -- which are written and verified in chunks.

Users of 8-inch drives have found that this big buffer technique is useful even when more than one drive is available. As Apparat points out, mini-disk drive heads remain engaged for an entire two-disk copy, even though only 10 sectors are being read or written in each chunk. With 8-inch drives, this system causes the heads to repeatedly engage and disengage with a lot of noise. The big buffer technique cuts this to a minimum.

Format 6. The final format available under NEWDOS 80's COPY command is a modified full-disk copy, with some important exceptions. This format permits copying by file, so that selected files can be transferred from the source disk to the destination disk. If the source disk is a system disk, and the user elects not to transfer the system files (except for BOOT/SYS, and DIR/SYS, which are transferred automatically), the destination disk will end up as a data diskette.

But first, an example to look at:

```
COPY,:0,TO,:1=60,01/04/81,CBF,CFWO,DDST=6
    ,DDGA=6
```

Many of the options available under Format 5 can be used, but I've left them off for clarity. In this example, the default value of Drive 1 has been bypassed in order to format it for 60 tracks. Because NFMT was not specified, the system first formats the destination disk (for 60 tracks in this case), and places BOOT/SYS and DIR/SYS. DDST has instructed NEWDOS 80 to place the first directory sector at the first sector of track 6 (instead of the customary track 17), and DDGA has allocated six granules to the directory (in place of the usual 2).

As you can see, Format 6 provides a great deal of flexibility in "building" new disk formats. The PDRIVE and SYSTEM commands in NEWDOS 80 allows you to change the default values for track numbers, directory locations, and directory granule allocation to be compatible with these custom configurations. But, non—NEWDOS 80 users will have a great deal of trouble accessing your files. You may or may not find this an advantage.

CBF, along with CFWO, are the Format 6 features that make it possible to copy by file. When CFWO (check file with operator) is invoked, each file will be presented, one at a time, for a "Y" or "N" copy decision, similarly to the opposite function, PURGE. The actual mechanics of the copying, and the order in which the directory entries and copies are made, are interesting, but not essential in understanding the results.

The bottom line is that the user can insert source and destination disks into a pair of drives, and copy only selected files (including system files) from one to the other. You can end up with a data disk or a full system disk, as you prefer. This format should be especially handy at users' group meetings, when members may wish to swap five or six programs they've written, which are contained on a single disk. The recipient can copy only those programs of interest, without having to KILL unwanted files duplicated through an indiscriminate full disk COPY.

Conclusion

You may have been waiting for one of your favorite magazines to carry an article that will completely explain all the advantages of NEWDOS 80. Perhaps this brief excursion into just some of the capabilities of just one of its commands will explain why no one article can summarize this fine DOS' feature.

In fact, as soon as someone out there fully digests the implications of NEWDOS 80's extended file handling capabilities, I'd like to read an article on that topic myself. (*So would we..Ed*) ●

MμMath

MμMath

Cameron C Brown

Now you can turn your computer into a mathematical genius. MuMath gives the TRS-80 user the sophistication that comes with years of training and practice. Integration, differentiation, complex numbers and polynomial expressions are handled as easily as plus and minus. The muMath package is supplied on a system disk for either a 32K or 48K Model I system. By entering from DOS the command "MATH48" (or "MATH32") you are into the world of higher mathematics.

Once the operating system is loaded, you may enter equations in a "calculator mode" and solve problems from arithmetic, algebra, trigonometry or calculus without having to program for a solution. Besides equation solving, the muMath system allows for simplification of expressions and computations of extremely large or small numbers (over 600 digit accuracyl).

The operating system uses a language called muSimp, and all routines used by muMath are in this language. MuSimp is a scaled down version of LISP that has been developed for the TRS-80. More on muSimp later. Before attempting to execute certain functions, muSimp routines are loaded from the disk. Only 48K users can take advantage of the calculus functions. All users have access to the additional source files for logarithmic or trigonometric simplification.

As a calculator, muMath works with rational arithmetic. All results are expressed as quotients, no decimals. Computational results from muMath are displayed with a preceeding @ symbol. For example, entering the command: 3/4 + (2/3)*(3/7); results in the answer @29/28. Expressions involving radicals are also reduced according to algebraic rules, e.g., (54)↑(1/3); results in @2↑(1/3) * 3. The power of muMath to handle large expressions is amazing. For example, 200! takes less than 6 seconds and is 375 digits long. It starts with 7886578673647 and ends with a 2 followed by 49 zeros.

Besides fancy arithmetic, base conversion, variable assignment, expansion factoring and evaluation of expressions, multi variate functions, and composition of functions are all possible. After loading the appropriate source files, you can even compute partial derivatives - DIF(X↑2 + 6*X*Y + Y↑2,Y) results in the answer @6*X+2*Y.

The ability to manipulate expressions and rational numbers made muMath a natural choice for our Math Department. To date, the student and teacher reactions have been mixed. The power and flexibility as a calculator has fascinated many students. As in any computer application, there is the added problem of learning syntax rules. We have found the muMath package very useful for students who already have a strong foundation in mathematical functions and notation. For others, an answer such as @3*X + (X↑2 +6)↑(1/2) is just a mystery. Students have pointed out that answers only do not explain process, and that the power-up and calling of disk routines confuses them. We find that students with a strong math background and a familiarity with computers can and do use muMath.

Teachers have noted that the package does lend itself to inquiry assignments; the kind where students are led through discovery problems to the recognition of a pattern or process. It has also been used to have students verify their solutions to assigned textbook problems. Through carefully guided assignments, muMath can be a very useful tool for the mathematics instructor.

MuSimp is the programming language in which the muMath routines are written. All muSimp routines are stored on the disk in ASCII files and it is possible to write your own programs in muSimp. MuSimp (an acronym for Microcomputer Structured IMPlementation language) is a surface language for a modified version of LISP. By using an editor such as Scripsit or Electric Pencil, you can create your own muSimp programs, store them as ASCII files and access them through the RDS (Read Disk Source) command of muMath.

A muSimp program is very easy to write. The language requires that programs be written in a structured, modular fashion. The syntax and control structures are few and you have access to all the functions used by muMath. Graphics commands - set, reset and point - for the TRS-80 are also included.

Here is a sample of muSimp. This program is a routine to determine if a number is prime:

```
FUNCTION PRIME(N,RESULT),
  RESULT:TRUE,
  X:2,
LOOP
  WHEN X = N EXIT,
  WHEN INTEGER (N/X)RESULT:FALSE EXIT,
  X:X+L,
ENDLOOP
ENDFUN $
```

This results in the following values PRIME(5); @TRUE and PRIME(346); @FALSE. The function PRIME(N) is now defined and can be referred to in any further calculations.

The muSimp language is also recursive. This allows you to use a function to calculate itself. As an example, the Fibonnaci sequence (1,1,2,3,5,8,13,21,34,...) can be defined by the following rules:

fib(0) = 1 fib(1) = 1
(first two terms are one)
fib(n) = fib(n-1) + fib(n-2)
(each term is the sum of the previous two terms)

Microsoft gives the following muSimp program to determine the nth term of the sequence:

```
FUNCTION FIB(N)
  WHEN N=0 OR N=1, 1 EXIT,
  FIB(N-1) + FIB(N-2),
ENDFUN $
```

The recursion occurs in this function because FIB(n) calls itself to evaluate itself.

Notice that we can now request composite functions such as PRIME(FIB(6)); @TRUE or PRIME(FIB(FIB(4))); @ FALSE. By carefully building muSimp modules, you can develop very elegant and extensive routines.

The language does have a drawback. It is slow to process source files and recursive routines. It is also very easy to exhaust the memory stack that is available for processing. Our students have found the language interesting and it is an excellent example of a structured, non-Basic language for programmers to investigate.

Included with the program package is a 76-page manual which covers all muMath functions. It is complete with examples and a subtle humor. The information on muSimp is very sketchy and there are few guidelines for learning how to program with it. We have also found that the muMath will not return to DOS as stated, but goes to MEMORY SIZE, a minor irritant.

If you are looking for a second language to study, a superior calculator or a tool for math education, consider muMath, it is fascinating. ●

MuMath a second look by

Terry Dettmann

For the utmost in flexibility, the kind demanded in artificial intelligence for example, a system like muMath from Microsoft is the best answer to your programming problems. As the accompanying article by Cameron Brown indicates, the system is loaded with flexibility, special abilities, and is generally fascinating.

This discussion will extend the last article on muMath with some more thoughts on what it can and cannot do and some ideas for programming with muSimp.

First though, for those who aren't familiar with it, let's learn a little about LISP.

What's a LISP???

LISP was invented in the 1950's by John McCarthy of MIT. LISP is an acronym for LISt Processor (but any college student who has ever studied LISP will be glad to tell you it really stands for Lots of Irritating, Silly Parentheses!). It was devised to handle *symbolic* manipulation rather than numerical calculations such as FORTRAN and Basic. As such, it has become the language most often used for programming artificial intelligence.

There is no reason that Basic cannot be used to do artificial intelligence programs. But LISP does them easier since the language gives the direct capability to the programmer to deal with symbolic structures such as sentences, algebraic equations and the like.

MuSimp is a LISP "surface language" that gives you much of the power of LISP but has a different series of instructions to accomplish its tasks.

LISP and muSimp work on data in the form of either *atoms* or *lists*. Each word in this article for example, is an atom. It's a combination of the basic characters in English put together in some meaningful combination.

We combine atoms to get lists using the rules of LISP and muSimp. Once we begin to look at an English sentence as a list and relate the properties of the language such as the presence of a subject and predicate, we can use manipulations in muSimp or LISP to break down the sentence into meaningful parts.

Using muSimp for example, we can identify the first atom in a list. If the list has the variable name X, then the first atom in the list is FIRST(X). We get similar functions for the SECOND and THIRD elements.

It is also possible for an element of a list to *be* a list. In this way, we can insert clauses in sentences as a list within a list. MuSimp can then be used to break down the logical structure of such a sentence.

Each of the functions in muSimp corresponds to a similar function in LISP. Generally, these aren't of the same name. If you are used to programming in LISP, it is necessary to learn how the syntax of muSimp differs.

For someone who is just learning LISP, either on their own or through a class in LISP, muSimp is a powerful way to simulate LISP, but the changed syntax will give you some problems. There are no books out on muSimp as such, though there are quite a few on LISP. It's hard to go from the books to practical programming in muSimp, even if you know what you are doing.

I have consulted the books (listed at the end of this article) on LISP. Some of them gave some interesting programs, none of which could be run directly. All LISP programs had to be adapted to work with muSimp.

The manual provided by Microsoft with the muMath package for TRS-80 is inadequate if you really want to do muSimp programming. You need to get the muMath/muSimp Reference Manual. This manual will give you insight as to what can and cannot be done with muSimp.

The manual was written by The Soft Warehouse of Hawaii, the original designers of the muMath/muSimp system. It is as the name implies, a reference manual, not an instruction book. If you are really serious though, you ought to have it.

Programming in muSimp

Cam Brown (see earlier article, this issue) has already given some insight into muSimp programming. The examples he gave are well worth trying. More exciting to me was the possibility of getting into artificial intelligence.

The problem in artificial intelligence is to make the machine handle symbolic structures like English and manipulate them in meaningful ways. To make an intelligent machine, capable of discussing in English instead of Basic - this is one of the fundamental aims.

A good example of where this is supposed to lead are the Robots in Asimov's Robot stories. We don't have the Positronic Brain though, just a TRS-80. It would be nice if it could understand a few simple sentences.

As a start, I took Tracton's book (Programmer's Guide to LISP) and tried to adapt a few of the LISP programs given in the back to muSimp. Each of the programs runs on the Control Data Corp computer I have available under LISP, so I thought it should be able to work with muSimp with changes.

I still haven't worked out all the differences, but light is beginning to show. Tracton has a program called Psychiatrist, written in LISP. I only have some parts of it running.

Psychiatrist carries on a very limited dialog with someone about a problem. It is a simplistic form of the popular ELIZA program. ELIZA, by the way, was originally written in LISP.

The reason for bringing up a not yet successful conversion is that the non-success has taught me some valuable lessons. First, where LISP uses DEFINE to create a function, muSimp uses FUNCTION. PRINT dosen't have the same effect in muSimp and it will not accept the abbreviation (') for QUOTE.

Somewhat more distressing is the problem of displaying prompt lines cleanly in muSimp. If you want to have a string printed, it always prints in format:

ATOM1 (ATOM2, ATOM3, ATOM4)

This is hardly what you want to use for programming prompts.

This is where the muSimp surface language falls down in its documentation. It does not clearly give you a correspondence between LISP and muSimp. Where muSimp is limited compared to LISP and how to program LISP function in muSimp. I am strictly an amateur with LISP. I've played with it, but never written a big application in it. For this reason, a muSimp correspondence would be especially useful.

What's it good for?

I see muMath/muSimp as a very flexible programming language for several types of people. First it will interest teachers. MuSimp in particular is a very structured programming language which forces good programming practices. This system is not for a beginning programming class however. Beginners seem to get lost too quickly in the complexities of the system if they have had no prior exposure to programming.

The second group I expect will like muMath/muSimp will be the students themselves. All the way up through simple

integral calculus (if you have 48K), a student can work with muMath to check answers to problems. It might even spark some work with more advanced function programming to answer the more difficult problems.

A third group who may be interested are the "Home Students", those people who learn by themselves. It can be used to supplement instruction they find elsewhere.

A final group I expect to see working with muMath/muSimp are the people interested in artificial intelligence and Robotics. MuSimp provides some of the power of LISP on a much smaller computer. An amazing amount can be done with it towards learning new techniques or trying some new ones.

MuMath/muSimp does have some limitations you should recognize. It deals strictly with integer numbers and fractions, no decimals. For many problems, this is serious. The system is also very slow. It will take you a long time for some things which involve long calculations. The reason is that muMath/muSimp does its manipulations by the same techniques you use to do them long hand.

I can't argue with 612 place accuracy on the numbers, but most serious applications require less accuracy on a wider range of numbers. In other words, you need real numbers and exponential notation.

If you are using the Model II, you don't have to wait if you really want to work with muMath. The full package is also available to run under CP/M from Microsoft. In this form, it can be run on the TRS-80 Model II with a CP/M operating system.

MuMath/muSimp is available from Microsoft Consumer Products, 400 108th Ave NE, Suite 200, Bellevue, WA 98004. It costs $74.95 for the TRS-80 Model I. The reference manual is available for $25.00. At present, the system is not available for the Model III. ●

References

Allen, J R, *Anatomy of LISP,* McGraw-Hill, 1978

Barden, William, *MuMath/MuSimp Instruction Book,* Microsoft Consumer Products, 1980

Graham, Neill, *Artificial Intelligence,* TAB Books, 1979

Organick, E E, Forsythe, A I and Plummer, R, *Programming Language Structures,* Academic Press, 1978

Siklossy, Laurent, *Let's Talk LISP,* Prentice-Hall, 1976

Tracton, Ken, *Programmer's Guide to LISP,* TAB Books, 1980

Weissman, Clark, *LISP 1.5 Primer,* Dickenson Publishing, 1967

MuMath/MuSimp-80 Symbolic Mathematics Systems Reference Manual, The Soft Warehouse, 1980

BYTE Magazine, *LISP Special Issue,* Vol 4, No 8, Byte Publications, August 1979

Note: This list is not complete. It is simply a place to get started in LISP, Artificial Intelligence and Symbolic Mathematics.

I'M A BELIEVER !!

"I Love it !!. . .It's really a incredible O/S. It' just great! Now I see why people who have seen it say they are now believers. I know I am."

LANCE MICKLUS

1) Large (8") drive support.
2) Double Sided drive support.
3) Double Density drive support.
4) 80 Track drive support.
*NOTE all above drives may be mixed on any one system and can be configured at Syssen time or during any Backup!
5) Winchester technology fixed drive support.
6) Supports any combination of the above drives up to a max. of 8 drives.
7) Supports double-speed processor clock modifications. (Archbold for example)
8) FASTER! ——— Improved overlay structure using ISAM accessing techniques improves loading times by up to 1400%.
9) General purpose output spoolers of a true, symbiont design provide simultaneous output and program execution without any user intervention.
10) Keyboard Type-Ahead feature permits you to enter keystrokes before your programs need them.
11) User definable keys, all 26 letters.
12) Built in Graphic string packer lets you enter graphic symbols into a BASIC program from the keyboard through the use of the (Clear) key. The (Clear) key is simply held down (just like the (Shift) keys) during other keystrokes and viola...graphics!
13) Dated files. —— All files are accompanied by the date of their last modification (creation or write).
14) Marked files. —— All files are accompanied by a 'mark' if they have been modified since they were last backed up. This permits the BACKUP utility to copy only those files which have actually been updated since a previous backup.
15) File transfer by class. Allows transferring of all files of a similar directory classification such as /CMD, /BAS, /PCL, etc.

VTOS 4.Ø

VTOS 4.Ø
Operating System Diskette with Operator's Guide
$99.95

VTOS 4.Ø
Master Reference Manual
$29.95

VTOS 4.Ø
Combination - 4.0 disk, Operator's Guide, and Master Reference Manual
$125.00

16) Built-in SYSTEM command contains lower case display driver, screen print, break key disable, blinking cursor, disk drive stepping rate and motor-on delay modifications, and more.
17) User may SYSGEN a custom VTOS system configuration containing special I/O drivers, device LINKing and ROUTEing, SPOOLing and DEBUG tasks, etc. which will be automatically loaded during the BOOT process without requiring a more lengthy AUTO and CHAIN procedure.
18) Non-BREAKable AUTO and CHAIN commands.
19) Wild-card DIRectory. Permits you to locate all files of a certain classification such as '/BAS'. Uniformly indicates file size in K (1024 bytes) regardless of drive type. "DIR D" would give you all your files that start with "D".
20) Dynamic file name defaults in APPEND, COPY, and RENAME commands allow you to specify only minimal information about file names.
21) COPY and APPEND commands execute up to 300% faster.
22) ALLOCate command for pre-allocation and non-releasibility of file space. File space will never shrink if this option used.
23) MEMORY command for directly setting upper memory limit.
24) Variable Length file support is incorporated which automatically blocks short user data records both within a sector and across sector boundaries thereby taking maximum advantage of disk file space.
25) No security disk needed to make backups or to run the system!
26) Though many O/S bear his design and code VTOS 4.Ø is the only Fully Aproved Operating System by Randy Cook! And it is FANTASTIC!
27) Endorsed by Scott Adams and Lance Micklus!

VTOS and VTOS 4.0 are registered trademarks of VIRTUAL TECHNOLOGY, INC. - Dallas, Texas 75234

Available from the following distributors or your local computer store. *DEALER INQUIRIES INVITED.*

5% Discount Just For Mentioning This Ad. (Valid month of this publication ONLY)

QUALITY SOFTWARE DISTRIBUTORS
11234 Park Central Pl Suite C
Dallas Texas 75230
(214) 692-1055
Micronet - 70130,203
SOURCE - TCC293

Circle 37
ADVENTURE INTERNATIONAL
Box 3435, Longwood, Fla. 32750
(305) 862-6917 - Voice
after 8:00 - same number
as FORUM 80. (SOURCE - TCC957)

SMALL BUSINESS SYSTEMS GROUP
Main St. and Lowell Road
Dunstable, Mass. 01827
(617) 692-3800 - Voice
(617) 692-3973 - FORUM 80
Micronet - 70310,236

Prices subject to change without notice

BASIC COMPILERS

A look at two currently available Compilers for the TRS-80.

T R Dettmann

Recently, two compilers came on the market for the TRS-80 Model II. They were produced by Microsoft and by Radio Shack.

You might think that one was identical with the other. It's logical anyway, since Microsoft supplied the Basic interpreter for the Model II. But it just isn't so!

As it turns out, Radio Shack has come out with a compiler that is not only different from Microsoft's, but in fact is incompatible with the interpreter Basicl Programs you already have running on your Model II under Basic cannot be compiled with Radio Shack's compiler without extensive changes.

Why not? I don't really know for sure. Money is a likely reason. But I won't go further; speculation will do us no good. But it seems illogical. However, Radio Shack does have the advantage of price when it comes to compilers.

Because it is so tempting, we thought it would be good to know how Radio Shack's and Microsoft's compilers compare. This is just a comparison, not a true evaluation or user report, since it is based on a limited use of the compilers.

What we will show is what we found in a close reading of the manual and some simple experiments. We have also tried to match corresponding Basic statements between the two compilers to see what works and what doesn't. Before we get into what they do, let's take a little side trip for those who might be wondering just what all the fuss is about.

What is a Compiler?

The American National Standard Vocabulary for Information Processing defines the word "compile" as: *To prepare a machine language program from a computer program written in another programming language by making use of the overall logic structure of the program, or generating more than one machine instruction for each symbolic statement, or both, ...*

In short, a "Compiler" translates programs from a high level language (such as Basic) directly to machine language. The machine language is then the program executed by the computer.

Why bother? Well, generally a compiled program may be anywhere from 2 to 30 times faster than an equivalent program in Basic. An interpreter, such as Basic supplied with your machine, translates each statement into understandable form every time it comes to the statement. Therefore, in a loop such as this one:

```
FOR I = 1 TO 1000
    Y(I) = X*Z(I)
NEXT I
```

The arithmetic statement in the middle has to be translated by the interpreter *every time through the loop*. This means that the interpreter must do the same thing to understand the statement 1000 times!

With the compiler, the statement is translated only once. This translation converts the expression in the loop directly to an equivalent machine language expression that the computer can work with. Since the translation is done when the program is compiled, it doesn't have to be done while the program is running.

Compilers are useful for more than just speeding up games. They also improve the response of business programs immensely. Imagine a mailing list sort improved in speed by 30 times. A half hour, compressed into one minute! It's not impossible - it is very possible with the right program and the right compiler.

Microsoft's Compiler

The Microsoft Compiler (available from Microsoft, 10800 NE 8th St., Bellevue, WA 98004, $400) was the first compiler available. It is based primarily on their Basic-80 interpreter, version 5.1. In fact, it is identical with their successful Basic compiler for CP/M systems.

The manual you get with the system is clear, but meant for the experienced programmer, not a novice. It is a reference to the language, not a text book. Using special compiler "switches" which tell the compiler what to do, you can set special conditions such as type of code generation, type of Basic, etc.

TRS-80 Basic is actually Microsoft's version 4.51 Basic interpreter. As such, the compiler objects to some of the things that are done with the language unless a special switch is set to tell it to accept 4.51 Basic.

Microsoft generates machine language code which can be "linked" together with other machine language programs into a single program file. In this way, you can write machine language routines with their M80 assembler, or other routines using their FORTRAN system, and using the "Link" package, you can bring them all together into a single program which runs on a single command.

Radio Shack sells their FORTRAN compiler for $299 and It is compatible with the BASIC compiler. It has gone through a number of stages of development. Some of them have had serious as well as some not so serious problems. Early users of the system were plagued by some of the errors. But the compiler has stabilized, and appears to be performing well.

Radio Shack's Compiler

The Radio Shack compiler is a relative to the Microsoft compiler only so far as they both work with the Basic

language. Past that, they are different. The Radio Shack compiler ($199 from any Radio Shack store) was written by Ryan-McFarland Corporation of Rolling Hills Estates, California. These are the same people who wrote the new COBOL compiler for the TRS-80 Model II.

RSBasic as it is called, is a development system with four modules. The first is a resident program which is always in memory, an editor which is called as needed to edit programs, a compiler which translates the programs, and a runtime package that provides special routines to handle input, output and a host of other tasks required by the Basic program.

The compiler has some very powerful features built in such as the ability to set and control breakpoints during execution of a program, do memory dumps, etc. The package also has a special runtime program called "RUNBASIC", which allows you to run compiled programs without the rest of the system or the Basic source code.

Radio Shack sells the RUNBASIC program separately, so you can have someone buy RUNBASIC from them and you sell them only your compiled program. This means that your source code is protected.

Even more significant to many is the fact that one of the most advanced data handling techniques, Indexed Sequential Access Methods (ISAM) is built into the package. This is also compatible with ISAM files created by the Radio Shack COBOL compiler. People familiar with this technique would certainly drool over its availability here.

The problem with the Radio Shack system is that it is incompatible with their Basic interpreter, and even with their own software. Since the interpreter was written by Microsoft and all the present software was developed with that interpreter, you have to do extensive modifications to programs in order to get them to compile.

All this makes me feel like I've been sold down the river by a bunch of slick, fast talking salesmen. If you buy the Radio Shack General Ledger program, you can't compile it without doing a whole lot of programming. Do I sound bitter? Yes I am. I've spent a lot of time since the Model II came out, developing programs for customers. During the development process, I was careful to maintain compatibility with the Microsoft compiler, never dreaming that Radio Shack would opt for something else. I have to admit that Radio Shack has noted in their catalog that their compiler is not compatible with the interpreter, but note or not, it just seems dumb.

Comparison of the Compilers

Table 1 is a comparative list made by listing all the Basic statements allowed under the Radio Shack compiler and then trying to match them with statements allowed with Microsoft's. Unmatched statements listed didn't seem to correspond between the languages.

We wrote some very short programs and compiled them using both compilers. They make for some interesting comparision. These programs were not an exhaustive test of the compilers. Table 2 gives the results of the tests in terms of the time it took for each program to run.

More extensive benchmarking is needed for the two compilers, but this will give an idea of their relative performance. The Radio Shack compiler was faster to compile a program for use. It took only 10 to 15 seconds for the sample programs. Microsoft's compiler required several minutes since each program had to be compiled by BASCOM and then linked using the L80 Linker in order to get a program that would run.

A Final Word

As with any test, there is some amount of opinion that comes out at the end. No matter how short the test, we tend to form opinions.

My final feeling is that at present I won't be buying the Radio Shack compiler. I can't see it as an advantage, since I have already put a lot of development effort into programs using the interpreter. For me, the only real advantage of Basic is being able to run a program under an interpreter to check it out quickly and surely. Once it's ready to go, I'd like to compile it for use.

Table 1

Comparison Chart of RSBASIC and BASCOM

RSBASIC	BASCOM
ABS	ABS
AND	AND
ASC	ASC
ATN	ATN
CALL	CALL
CHAIN	****
CHR$	CHR$
CLOSE	CLOSE
COM	****
COS	COS
CRT	****
CRTG	****
CRTI$	****
CRTR	****
CRTX	POS
CRTY	POS
CVD	CVD
CVI	CVI
DATA	DATA
DATE$	****
DEF	DEF
DELETE	****
DIG	****
DIM	DIM
END	END
EOF	EOF
ERR	ERR
ERROR	ERROR
EXP	EXP
EXP10	****
EXT	DEFUSR
FOR NEXT	FOR NEXT
GOSUB	GOSUB
GOTO	GOTO
HEX$	HEX$
HVL	****
IF-THEN-ELSE	IF-THEN-ELSE
INKEY$	INKEY$
INPUT	INPUT
INPUT USING	****
INPUT$	****
INT	INT
INTEGER	DEFINT
KILL	KILL
LEN	LEN
LINE INPUT	LINE INPUT
LOG	LOG
LOG10	****
LPRINT	LPRINT
LPRINT USING	LPRINT USING

Comparison Chart of RSBASIC and BASCOM

RSBASIC	BASCOM
NOT	NOT
ON BREAK GOTO	****
ON ERROR GOTO	ON ERROR GOTO
ON-GOTO	ON-GOTO
ON-GOSUB	ON-GOSUB
OPEN	OPEN
OR	OR
POS	INSTR
PRINT	PRINT
PRINT USING	PRINT USING
RANDOMIZE	RANDOMIZE
READ	READ
REAL	DEFSNG
REM	REM
RESET BREAK	****
RESET ERROR	ON ERROR GOTO
RESET GOSUB	****
RESTORE	RESTORE
RESUME	RESUME
RETURN	RETURN
RND	RND
SEG$	MID$
SGN	SGN
SIN	SIN
SQR	SQR
STOP	STOP
STR$	STR$
STRING	DEFSTR
STRING$	STRING$
SUB	****
SUBEND	RETURN
SWAP	SWAP
SYSTEM	****
TAB	TAB
TAN	TAN
TIME$	****
VAL	VAL
WRITE	WRITE
XOR	XOR

Note: 1 - This table was prepared by listing the RSBASIC commands from the command table in the manual and trying to find a command in Microsoft's Basic reference manual that corresponded with it. It is not based on experience with RSBASIC and so some entries may be due to misinterpretation of the documentation. 2 - A comparison of the statements available doesn't tell everything about the comparison. As illustrated in the test programs, even though RSBASIC and BASCOM use the OPEN statement for file handling, they use it differently. 3 - The following statements available under BASCOM were not matched with any from RSBASIC: ERL FIELD GET CDBL CINT CSNG CVS FIX FRE LSET RSET NAME NULL OUT POKE PEEK PUT INP LEFT$ RIGHT$ LOC LPOS MKI$ MKD$ MKS$ OCT$ IMP EQV SPACE$ WAIT WIDTH SPC USR VARPTR

This is not a list of statements applicable to the interpreter, just to the compiler from Microsoft.

TABLE 2

Comparison of the Compilers

PROGRAM	RSBASIC	BASCOM	INTERPRETER
Test 1	0:55	0:40	0:59
Test 2	2:03	2:37	3:18

NOTE: 1 - These tests were timed with a stop watch from the ENTER before each loop to the appearance of STOP on the screen. 2 - RSBASIC and BASCOM did not accept ENTER as the only character for the INPUT statements at the beginning of each program. A single character had to be typed in at each START prompt. 3 - All comparisons were run on DOS 2.0 ●

Protect Your Fields

Bill Stevener
Las Vegas, NV

For the Model I TRS-80

If you are looking for a professional data entry routine, or you like new approaches to old problems, or want to make your terminal look like one of the biggies - this article is for you.

Background

My part-time hobby is developing software for small businesses using the TRS-80. One of these companies compares existing life insurance policies to proposed policies, and "writes" a proposed policy package for the insurance sales agent. When proposing life insurance replacement policies, a somewhat standardized replacement form must be completed and given to the customer (see Figure 1). As you can see, existing and proposed information is needed for this form. My program generates all the proposed information, but I need a way to enter the existing information, so I used many INPUT statements, FOR-NEXT loops, etc., with appropriate prompts. All 55 data items were entered, but as will always happen, mistakes were made, the operator was not always sure what he entered 29 items ago, and sometimes hit ENTER one time too many, bypassing an item. Finally, one day my customer jokingly asked me if he could use the up-arrow to back up and correct a mistake.

Designing the Routine

That question got me thinking. I have seen and used protected and unprotected fields on some terminals. You see a blank form on the screen and use the arrow keys to position the cursor to the correct field. Since I enjoy a good programming challenge and wanted to impress my customer, I worked up the following routine which does this for the TRS-80.

Using a TRS-80 video worksheet, I designed the screen (see Figure 2). I used XXX's to identify alphanumeric fields and 999's to identify strictly numeric fields. I numbered the fields, so I could keep track of where they were. I sent this modification to my customer, only telling him to use the arrow keys to move the cursor, and even he was able to take it from there!

Having completed the worksheet to my liking, I started coding. I won't bore you with all the details but will provide a copy of the final output during data entry (Figure 3), a listing of the routine, and briefly explain how it works.

Program Description

Lines 10-80. I needed an array (ZA(1,54)) to hold fielding information. ZA(0,n) tells me the video memory location for field n, and ZA(1,n) tells the length of field n (negative if numeric data). I also needed to get up some strings so I could save memory when printing the screen. Finally, I filled the ZA array with appropriate data (other arrays were used for data storage).

Lines 100 - 120. House-keeping chores. Clear the screen, increment the policy counter (I), print the screen, set the field offset (J), set the field counter (IC), set the previous character holder (IX) and eliminate any stray keyboard inputs (A$=INKEY$).

Lines 130 - 140. Replace the previous character, update the field counter (IC), store the previous character (IX), position the block cursor (CHR$(143), and strobe for input.

Lines 150 - 210. If the character was an arrow or ENTER, set the field offset (J) and move to the new field. If the character was a (SHIFT X), restore the previous character and process the data. If the character was anything else, display this character and go to a routine which will accept and display additional characters until either ENTER is pressed or the field is full. Upon return, update the field offset (J) and start a new field.

Lines 220 - 280. Set the character counter (IJ), accept and display characters individually via the INKEY$ function until ENTER is pressed or the character counter (IJ) is equal to the absolute value of the field length (ABS(ZA(1,IC))). The routine will accept all characters whose ASCII values are between 32 and 90 (blank and normal z), inclusive, unless the field is negative, indicating numeric data only, in which case 0-9 and "." are the only allowable characters. If the input is terminated with ENTER, an "!" is a flag used to signify the end of the field. *Note that by using the ! as an end-of-field marker, it may not be used as a part of the data entry field.*

Lines 290 - 350. If the first character of the company field is a blank, !, or -, that signals that the final company has been entered and the program can go about its other chores. Otherwise, call a routine which transfers each field, one at a time, into A$. If A$ returns empty, proceed to the next field. Otherwise test the counter to determine which field is in A$, and fill the appropriate data item or variable. When all fields are completed, the routine proceeds to accept the next company's data.

Lines 360 - 390. Set the video memory pointer (IC) to the proper field location. If the first character is a -, blank or !, return with A$ empty. Otherwise scan the field for the length of the field, or until an ! is found, whichever comes first. Once the actual field length is found (JI), A$ is initially set to 12 blanks (my maximum field length), then POKEd with PEEKs from the video memory.

```
1 REM
2 REM          DATA ENTRY ROUTINE
3 REM
4 REM          BY
5 REM          WILLIAM M. STEVENER
6 REM          4342 PARAMOUNT STREET
7 REM          LAS VEGAS, NV 89110
9 '    DEFINE AND SET UP THE VARIABLES
10 CLEAR1000:DEFINTI,J,Z:I=0:J=0:Z=0:DIMZA(1,54):POKE16553,255:
     IM=7
19 '    ARRAYS USED IN MY PROGRAM, YOU MAY USE YOUR OWN
20 DIMDD(IM,1,7),DD$(IM,2,7),E1$(IM),E2$(IM),E3$(IM),EC$(IM),EG
     $(IM),EP$(IM),ET$(IM),IA(IM),LR(IM),TC(IM),GC(IM,5),CC(IM,2)
     ,DI$(IM),IS(IM),DV(IM,5)
```

```
29 '   SET UP MY STRINGS TO SAVE MEMORY WHEN PRINTING THE FORM
30 A1$="----------  R1 ----------  R2 ----------  R3 ----------
    ":A2$="     AD ----------  OP ----------  WP ----------  D
    I ----------":A3$="NOW -------- +1 YR -------- +5 YR -------
    - +10YR --------"
40 A4$="  +20YR --------  @ 65 --------"
49 '   READ IN THE FORM FIELDS
50 FORI=0TO54:READZA(0,I),ZA(1,I):ZA(0,I)=ZA(0,I)+15360:NEXT
60 DATA30,12,52,11,72,12,94,12,116,11,136,12,158,12,180,11,200,
    10,215,10,230,10,245,10,264,10,279,10,294,10,309,10
70 DATA328,10,343,10,358,10,373,10,392,10,407,10,422,10,437,10,
    456,-10,471,-10,486,-10,501,-10,520,-10,535,-10,550,-10,565,
    -10
80 DATA584,10,599,10,614,10,629,10,648,10,663,10,678,10,693,10,
    712,-8,727,-8,742,-8,757,-8,776,-8,791,-8,807,-4,819,-2,827,
    -2,840,-8,855,-8,870,-8,885,-8,904,-8,919,-8
89 '   INITIALIZE THESE ARRAYS
90 FORJ=0TOIM:FORI=0TO5:LETGC(J,I)=Z:LETDV(J,I)=Z:NEXTI:FORI=0T
    07:LETDD(J,0,I)=Z:LETDD(J,1,I)=Z:NEXTI,J:I=0
99 '   PRINT THE FORM ON THE SCREEN
100 CLS:I=I+1:PRINT"EXISTING POLICY #";I;"  COMPANY ----------
    -  NUMBER  ----------":PRINT"GENERIC ----------  BASIC P
    ----------  ISU DTE ----------":PRINT"RIDER 1 ----------
    -  RIDER 2 ----------  RIDER 3 ----------"
110 PRINT"PREM BA ";A1$:PRINTA2$:PRINT"PAY TO  ";A1$:PRINTA2$:P
    RINT"D.B.    ";A1$:PRINT"      AD ---------- CDB ---------- C
    AD ---------- CPL ----------":PRINT"A.B.C.  ";A1$:PRINTA2$:P
    RINT"CV  ";A3$
120 PRINTA4$;"  LN RT ----  SCALE --  AGE --":PRINT"DIV ";A3$:P
    RINTA4$:PRINT"                **  PRESS (SHIFT) X WHEN FINISHE
    D **";:J=0:IC=0:IX=45:A$=INKEY$
129 '   INITIAL HOUSE-KEEPING CHORES
130 POKEZA(0,IC),IX:IC=IC+J:IX=PEEK(ZA(0,IC)):POKEZA(0,IC),143
139 '   STROBE FOR INPUT
140 A$=INKEY$:IFA$=""THEN140
149 '   UP ARROW
150 IFA$=CHR$(91)THENIFIC(3THEN140ELSEIFIC(8THENJ=-3:GOTO130:EL
    SEJ=-4:GOTO130
159 '   DOWN ARROW
160 IFA$=CHR$(10)THENIFIC)49THEN140ELSEIFIC)7THENJ=4:GOTO130:EL
    SEJ=3:GOTO130
169 '   RIGHT ARROW, OR ENTER
170 IFA$=CHR$(9)ORA$=CHR$(13)THENIFIC)53THEN140ELSEJ=1:GOTO130
179 '  LEFT ARROW
180 IFA$=CHR$(8)THENIFIC(1THEN140ELSEJ=-1:GOTO130
189 '   SHIFT X
190 IFA$="x"THENPOKEZA(0,IC),IX:GOTO290
199 '   ANY OTHER CHARACTER, GET REST OF FIELD
200 GOSUB220:IFIC(54THENJ=1ELSEJ=-54
209 '   GET NEXT FIELD
210 IC=IC+J:IX=PEEK(ZA(0,IC)):POKEZA(0,IC),143:GOTO140
217 '
218 '
219 '   ROUTINE TO GET CHARACTERS FROM THE KEYBOARD (INKEY$)
220 IJ=0:GOTO270
230 IX=PEEK(ZA(0,IC)+IJ):POKEZA(0,IC)+IJ,143
240 A$=INKEY$:IFA$=""THEN240
249 '   IF BACK ARROW (ERASE)
250 IFA$=CHR$(8)THENIFIJ=0THEN240ELSEPOKEZA(0,IC)+IJ,IX:IJ=IJ-1
    :GOTO230
259 '   IF ENTER, SET !, BLANK REST OF FIELD
260 IFA$=CHR$(13)THENPOKEZA(0,IC)+IJ,33:FORJI=IJ+1TOABS(ZA(1,IC
    )):POKEZA(0,IC)+JI,32:NEXT:RETURN
269 '   IS CHARACTER ACCEPTABLE?
270 IFA$(CHR$(32)ORA$)"Z"ORZA(1,IC)(0AND(A$("0"ORA$)"9")ANDA$()
    "."THEN240
279 '   DISPLAY CHARACTER, CHECK TO SEE IF THE FIELD IS FULL
280 POKEZA(0,IC)+IJ,ASC(A$):IJ=IJ+1:IFIJ=ABS(ZA(1,IC))THENRETUR
    NELSE230
288 '
289 '   SHIFT X WAS GIVEN, NOW ASSIGN PROPER VARIABLES
290 PRINT@960,"                              ***     WORKING HARD     ***
    ";:IFPEEK(ZA(0,0))=32ORPEEK(ZA(0,0))=33ORPEEK(ZA(0,0))=45
    THENIE=I-1:CLS:GOTO400
300 FORJ=0TO54:GOSUB360:IFA$=""THEN340ELSEIFJ=0THENEC$(I)=A$:GO
    TO340:ELSEIFJ=1THENEP$(I)=A$:GOTO340:ELSEIFJ=2THENEG$(I)=A$:
    GOTO340:ELSEIFJ=3THENET$(I)=A$:GOTO340:ELSEIFJ=4THENDI$(I)=A
    $:GOTO340
310 IFJ=5THENE1$(I)=A$:GOTO340:ELSEIFJ=6THENE2$(I)=A$:GOTO340:E
    LSEIFJ=7THENE3$(I)=A$:GOTO340:ELSEIFJ(16THENDD$(I,0,J-8)=A$:
    GOTO340:ELSEIFJ(24THENDD$(I,1,J-16)=A$:GOTO340
320 IFJ(29THENDD(I,1,J-24)=VAL(A$):GOTO340:ELSEIFJ(32THENCC(I,J
    -29)=VAL(A$):GOTO340:ELSEIFJ(40THENDD$(I,2,J-32)=A$:GOTO340:
    ELSEIFJ(46THENGC(I,J-40)=VAL(A$):GOTO340:ELSEIFJ=46THENLR(I)
    =VAL(A$):GOTO340
330 IFJ=47THENIS(I)=VAL(A$):GOTO340:ELSEIFJ=48THENIA(I)=VAL(A$)
    :GOTO340:ELSEDV(I,J-49)=VAL(A$)
339 '   COMPUTE THE PREMIUM TOTAL AND DISPLAY IT
340 NEXTJ:TC(I)=0:FORJ=0TO7:TC(I)=TC(I)+VAL(DD$(I,0,J)):NEXT:PR
    INT@929,"TOTAL PREM =";:PRINTUSING"$$####.##";TC(I):PRINT
    "                    PRESS ENTER TO CONTINUE";:A$=INKEY$
349 '   WAIT FOR THE USER TO PRESS ANY KEY BEFORE CONTINUING
350 A$=INKEY$:IFA$=""THEN350ELSE100
357 '
358 '   ROUTINE TO PUT EACH FIELD FROM THE SCREEN INTO A$
360 IC=ZA(0,J):IFPEEK(IC)=45ORPEEK(IC)=32ORPEEK(IC)=33THENA$=""
    :RETURN
369 '   CONTINUE FOR LENGTH OF FIELD, OR ! IS FOUND.  JI=LENGTH
370 FORIJ=0TOABS(ZA(1,J))-1:IFPEEK(IC+IJ)()33THENJI=IJ+1:NEXT:E
    LSEIFIJ(=ABS(ZA(1,J))-1THENIJ=15:NEXT
379 '   INITIALIZE A$, POKE IT WITH THE FIELD CONTENTS
380 A$=STRING$(12,32):POKEVARPTR(A$),JI:MM=PEEK(VARPTR(A$)+2)*2
    56+PEEK(VARPTR(A$)+1):IFMM)32767THENMM=MM-65536
390 FORIJ=0TOJI-1:POKEMM+IJ,PEEK(IC+IJ):NEXT:RETURN
398 '
399 '   ALL DONE
400 CLS:PRINT:PRINT:PRINT:PRINTCHR$(23):PRINTTAB(10);"FINISHED"
410 GOTO410
```

Name of Proposed Insured Bill Stevener Address 4342 Paramount St Birth Date 5/24/80

GENERAL INFORMATION	EXISTING INSURANCE	PROPOSED INSURANCE
Name of Company	INDEPENDENT	
Policy Number	987654321	
Basic Policy Generic Name	Whole Life	
Name of Basic Policy	Best-West	
Rider 1, Generic Name	Family	
Rider 2, Generic Name	Child	
Rider 3, Generic Name		
Issue Age	29	
Date of Issue	20 Jan 76	
Contestable Period Expires	2 YRS	
Suicide Clause Expires	2 YRS	

PREMIUM DATA/DEATH BENEFITS	Premium Mode Amount	Age Payable To	Death Benefit	Age Benefit Ceases	Premium Mode Amount	Age Payable To	Death Benefit	Age Benefit Ceases
Basic Policy	$1034.24	LIFE	$75000	DEATH	$		$	
Rider 1	$ 40.00	55	$10,000	55	$		$	
Rider 2	$ INCL	21	$ 5,000	21	$		$	
Rider 3	$		$		$		$	
Accidental Death Benefit	$		$		$		$	
Option to Purchase Additional Insurance	$	(Opt. Ages)			$	(Opt. Ages)		
Waiver of Premium Benefit	$ 23.74	60	xxxxx	60	$	xxxxx		
Disability Income Benefit	$		xxxxx		$	xxxxx		
			(Mo. Income $)			(Mo. Income $)		
Total Current Premium	$1097.98				$			

CASH VALUES/DIVIDENDS	*Guaranteed Cash Value	*Dividends	Guaranteed Cash Value	*Dividends
Currently (last policy anniversary)	$ 10	$ 10	$	$
1 year hence	$ 20	$ 20	$	$
5 years hence	$ 30	$ 30	$	$
10 years hence	$ 40	$ 40	$	$
At Age 65	$ 60	$ 60	$	$

*Current Cash Value of Div Adds ---
*Current Death Benefit of Div. Adds $
*Current Accum. Div. $
*Current Policy Loan $

Maximum Policy Loan Interest Rate 7.5 %
*Dividends are based on the current (1976) scale

Maximum Policy Loan Interest Rate ____ %
Dividends are based on the current (19____)

*Dividends, policy loan and certain guaranteed cash value information concerning your existing insurance may not be known to our agent. Dividends are not guaranteed. However, they may materially reduce the cost of insurance and are an important factor to consider. Thus, if dividends or other figures have been omitted from this Disclosure Statement you should not reach a final decision to replace your existing insurance until you have them. You may obtain the omitted figures from the company that issued your existing policy. We will notify that company of your intent to replace your existing policy.

Figure 1

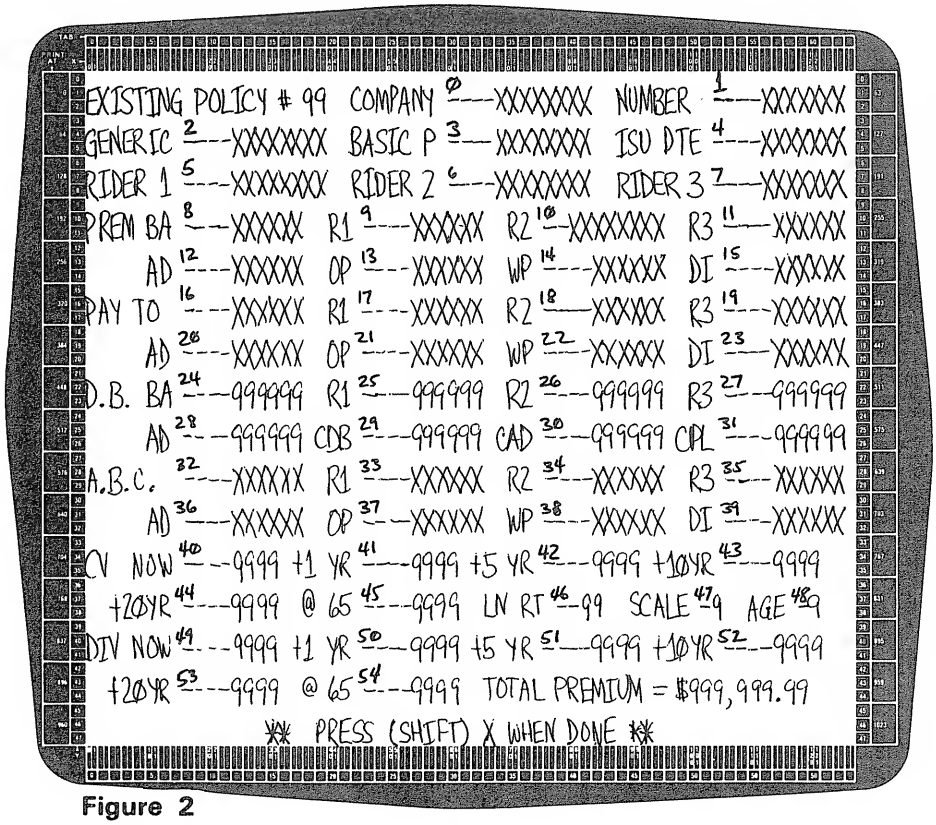


Figure 2

```
EXISTING POLICY # 2  COMPANY INDEPENDENT'  NUMBER  987654321'
GENERIC WHOLE LIFE'  BASIC P BEST-WEST'    ISU DTE 20 JAN 76'
RIDER 1 FAMILY'      RIDER 2 CHILD'        RIDER 3 ----------
PREM BA 1034.24'     R1 40'  R2 INCLUDED'  R3 ----------
        AD ----------        OP ----------  WP 23.74'  DI ----------
PAY TO  LIFE'        R1 55'  R2 ----------  R3 ----------
        AD ----------        OP ----------  WP 60'     DI ----------
D.B.    AD 75000'    R1 10000'  R2 5000'    R3 ----------
                     CDB ----------  CAD ----------  CPL ----------
A.B.C.  DEATH'       R1 55'  R2 21'         R3 ----------
        AD ----------        OP ----------  WP 60'     DI ----------
CV  NOW 10'    +1 YR 20'   +5 YR 30'   +10YR 40'
    +20YR 50'  @ 65 60'    LN RT 7.5'  SCALE 76  AGE 29
DIV NOW 10'    +1 YR 20'   +5 YR 30'   +10YR 40'
    +20YR 50'  @ 65 60'    TOTAL PREM = $1,097.98
                           PRESS ENTER TO CONTINUE
```

Figure 3

User Modifications

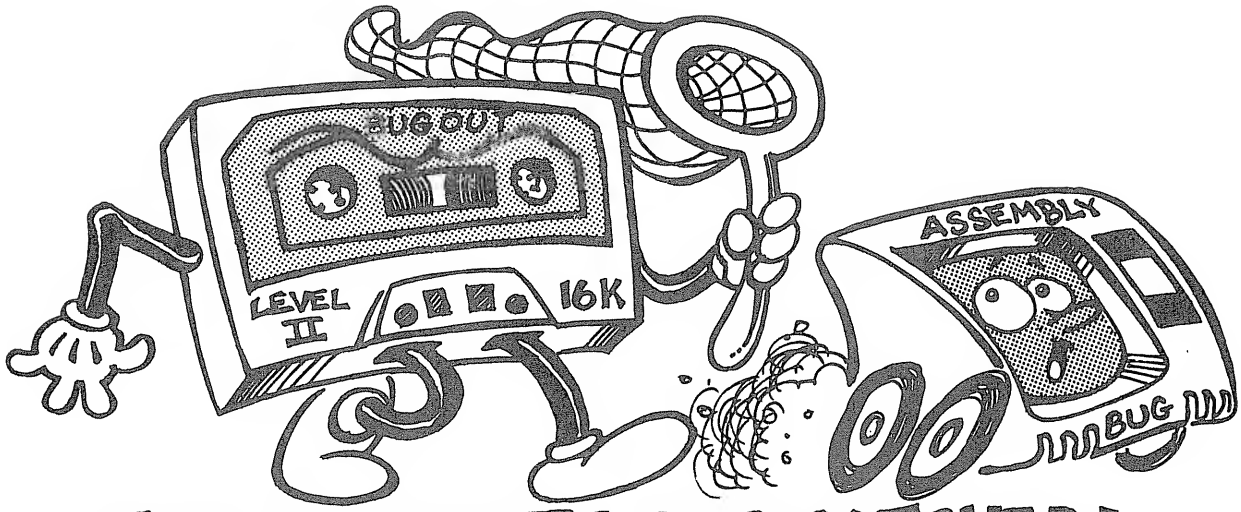
I would be surprised if anyone could use this program as is, so here are some ways you can modify it to fit your needs. Note that I have listed the routine in such a manner that if you want to type it in, you may use the AUTO function, since all necessary lines are on line numbers in increments of 10, and all remarks are on other line numbers.

First, you must design your own video display format. Once you have that, it helps to number the fields. Redimension the ZA array to fit your number of fields by changing the 54 in line 10 to one less than the number you need (remember you are using field #0). Also change the 54 in the FOR-NEXT loops in lines 50 and 300 to match your array dimension. Next replace the data statements with your own. The format is: DATA PRINT@#, length, PRINT@#, length, etc. Note that the FOR-NEXT loop will add 15360 to each PRINT@#, resulting in the actual video memory location for that field. Also remember to enter a negative number for each numeric field. Replace lines 100-120 to print your form as it appears on your video worksheet. Change lines 150-180 to indicate how you want to move the cursor when the arrow keys are pressed. Change lines 300-340 to assign variables properly. Finally, change the 12 in line 380 to conform to your maximum field length.

If you want to change the initial dashes to something else, change the 45 in lines 120, 290 and 360 to the appropriate ASCII code for the character you want. If you want to change the ! as end-of-field marker, change the 33 in lines 260, 290, 360 and 370 to the ASCII character you want.

Conclusion

This routine has added a professional touch to my program, as well as greatly simplifying data entry. The user can view the entire form, make changes where necessary, and ENTER it all at once.



THE ULTIMATE BUG CATCHER!

BUGOUT — A PROGRAM, TEXTBOOK, AND REVOLUTION!

THE BUGOUT PROGRAM is the most powerful Z-80 MONITOR/DISASSEMBLER/DEBUGGER and GENERAL PURPOSE UTILITY PACKAGE available! BUGOUT is so versatile, beginners can learn Assembly Language programming in record time. BUGOUT is so powerful, experts will refuse to program without it! Here are only a few of BUGOUT'S 90 individual functions.

- **Interpretive Traces**
- **Instant Breakpoints**
- **Address Breakpoints**
- **Register Breakpoints**
- **High Speed Traces**
- **ROM Analysis Routine**
- **1 Column Disassembly**
- **2 Column Disassembly**
- **Single Step in RAM**
- **Single Step in ROM**
- **Fast Patch Routine**
- **System FMT Tape I/O**
- **Base Program in 7K**
- **Bugout Relocator + 1K**
- **Video Save/Recall + 1K**
- **Extensive Memory Exam**

THE BUGOUT TEXTBOOK will answer questions other Assembly Language courses overlook, or cannot easily answer. There is no other course like it anywhere! The BUGOUT manual is written with the complete beginner in mind, yet provides interesting and valuable ideas for all programmers! Experts can easily skip what they already know and get down to important things....like squashing bugs in record time!

THE BUGOUT REVOLUTION, program and textbook combined, makes learning Assembly Language easy. New concepts are immediately brought to life on your video display! You can actually "see" how every machine code functions right before your eyes! You can enter BASIC language statements and single step trace them...all the way through the ROMS. You can link machine code routines to your BASIC programs more EASILY and EFFICIENTLY than you ever dreamed possible, and BUGOUT will remain a POWERFUL WORKING TOOL long after you've become an Assembly Language Expert!

HONEST TO BUGOUT — We couldn't begin to tell you everything about BUGOUT on one page, so we prepared a special information package for your inspection. Better yet, order BUGOUT today while it is still introductory priced at $24.95. We think this package is worth considerably more, and anticipate a price adjustment in the future. Order today and save...money, of course, but more important, your valuable time. BUGOUT will pay for itself immediately by saving you countless hours learning, writing, and debugging your own Assembly Language programs!.

RESERVE YOUR COPY NOW! BUGOUT BELONGS IN EVERY PROGRAM LIBRARY!

BUGOUT SWEEPSTAKES

Win $1,000 Cash

ENTRY DEADLINE — October 15, 1981

DRAWING — October 31, 1981

No purchase necessary to win.
Complete details provided to every entrant.

THE SOFTWARE PLANTATION, INC.,
P.O. Box 44623, Tacoma, WA 98444
(206) 531-1506

SWEEPSTAKES ENTRY FORM

Complete sweepstakes rules and procedures enclosed with BUGOUT or BUGOUT INFORMATION PACKAGE.
No purchase necessary. Void where prohibited by law.

TO PURCHASE BUGOUT, complete and mail to:
THE SOFTWARE PLANTATION, INC.,
P.O. Box 44623, Tacoma, WA 98444

____Check ____Money Order ____Master Card ____Visa
Card No. _____ Expiration date _____
Washington residents add 5.3% sales tax

____ Please rush BUGOUT @ $24.95
____ Please rush BUGOUT INFORMATION
____ Please enter my name for the drawing

Name _____
Address _____
City _____ State _____ Zip _____

Circle 39

PANATTONI'S PANACEA

Memory Expansion 32/48K

Larry Panattoni

"I'll trade you a copy of Maze which I typed from 80-U.S.", Jerry said, "for a copy of Startrek III.4 which you typed from Softside."

I thought for a moment, then replied "but Startrek takes about 20K, with all its remarks, and you have only 16K."

So for the next couple of hours we fumbled with two different remark removing programs, in an attempt to confine the program within 16K. Finally, we removed the remarks by hand.

"Jerry", I said, "what you need is more memory. I realize you are not about to spend $300 for an expansion interface because without disks or printer you don't need all the features the interface provides. But, if you were to build only the portion which enables you to expand your memory, it would cost less than $23, including a $9 power supply."

"But don't misunderstand - the $300 I quoted you as the cost does not include any memory chips. Neither does the $23.00 one you build yourself. However, memory chips are plentiful and prices are low."

This brought about his complete interest, and he inquired how he could get information on this $23 project. Here is what I told him:

TRS-80 Memory Layout

The '80 is designed to add memory in 16K blocks. The first 16K RAM is physically located within the keyboard unit, and additional 16K blocks, to 32K and 48K, may be added in the expansion interface. Figures 1 and 2 show the actual wiring and component designations used in the Radio Shack interface. Figure 2 shows the 32K and 48K blocks. Z9 through Z16 is the 32K block, while Z1 through Z8 are the 48K block.

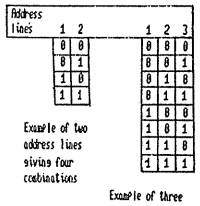
If you had 16K in the keyboard and purchased an expansion interface, then to up-grade to 32K you would have to obtain a set of eight 4116 dynamic memory RAM chips and install them in sockets Z9 through Z16. Later, to up-grade to 48K, you would have to install another set of eight 4116 RAM chips in sockets Z1 through Z8. Each 4116 chip is a 1 X 16 memory unit. Therefore, eight are needed, one for each of the 8 bits (D0 through D7), which comprise the 8 bit byte the TRS-80 uses.

Memory Addressing via Multiplexing

Chart 1 shows that 14 address leads are needed to address 16 different memory locations. However, from Figure 2 you can see there are only seven address inputs, for each of the 4116's. The method used to overcome this shortage of inputs is called "multiplexing". This means that the 1st of seven address signals are sent into the 4116's, then the 2nd seven address signals are switched onto the same seven input leads of the 4116 memory units, totaling the 14 address signals required. The internal memory unit circuitry remembers and places the two groups of seven address signals together. One group is identified as the Row Address Signals (RAS) and the other as the Column Address Signals (CAS).

This is better understood by observing the switching units Z17, Z18 of Figure 1. The four output leads of Z18 and the three output leads of Z17 go directly to the seven address input pins of the memory units in Figure 2. During the first phase of address multiplexing A0 through A3 from the interface bus, is switched through Z18 to the four outputs A0 through A3. A4 through A6 are switched through Z17 to the three outputs A4 through A6. At the proper time, the signal on the multiplex (MUX) lead is changed from low to high, causing the condition explained above to no longer exist, but instead, A7 through A10 of the interface bus, are now switched through Z18 to leads A3 respectively.

Chart 2 shows the various phases of the signals which control this operation. At the top is the 10.6445 Mhz clock from which



Nr of address lines.	Binary Representation of each Column													Nr of Combinations
	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	
1														X	= 2
2													X	X	= 4
3												X	X	X	= 8
4											X	X	X	X	= 16
5										X	X	X	X	X	= 32
6									X	X	X	X	X	X	= 64
7								X	X	X	X	X	X	X	= 128
8							X	X	X	X	X	X	X	X	= 256
9						X	X	X	X	X	X	X	X	X	= 512
10					X	X	X	X	X	X	X	X	X	X	= 1024
11				X	X	X	X	X	X	X	X	X	X	X	= 2048
12			X	X	X	X	X	X	X	X	X	X	X	X	= 4096
13		X	X	X	X	X	X	X	X	X	X	X	X	X	= 8192
14	X	X	X	X	X	X	X	X	X	X	X	X	X	X	= 16,384

Chart #1

The large chart above shows that 14 address lines are needed to access 16K different memory locations; as well as various other combinations. Examples of how this is done, is shown for a two address line and a three address line

HARDWARE
MEMORY EXPANSION 32K - 48K

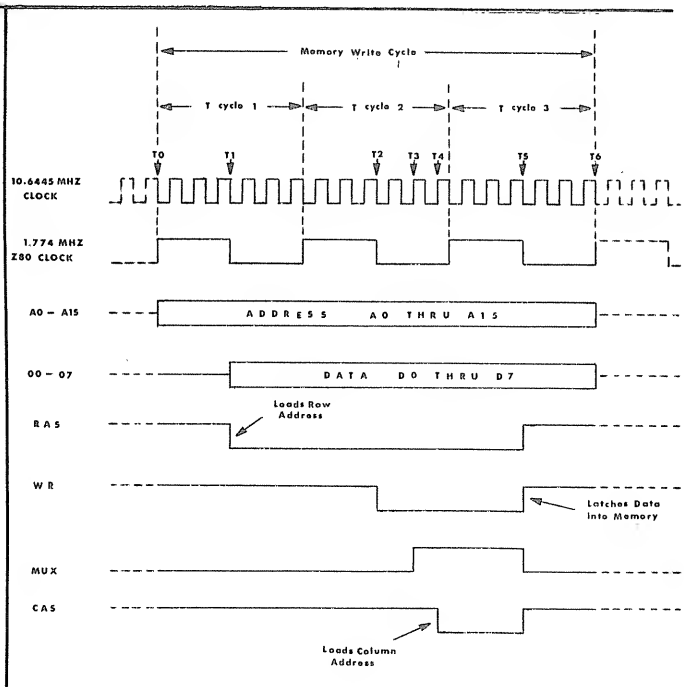


Chart #2

The timing waveforms of the various signals used during a "Write" operation is shown above. This would be the CPU outputting data to be stored in memory -- such as the command "LD (HL),A"

HARDWARE
MEMORY EXPANSION 32K - 48K

ine '80 derives all of its timing functions. The 1.774 Mhz clock, shown next, is used by the Z80 Central Processing Unit (CPU).

Let's say the CPU wants to write (WR) data to a certain memory location. The 16 bit address is placed on the address bus, AO through A15 at time zero (TO). After allowing time for the address signals to stabilize; the row address signal (RAS) is brought low at time T1, causing the memory units to accept the first seven (ROW) address signals. Also at time T1, data is placed on the data bus, D0 through D7 by the CPU. Then at time T2, the WR signal is brought low. (If we had been performing a read, the RD signal would have been brought low instead of the WR).

Next, at time T3, the multiplex signal MUX is brought high. This switches the inputs of Z17 and Z18 from the 1st seven row address signals, A0 through A6 to the 2nd seven column address signals A7 through A13. Again, after allowing time for these signals to stabilize, the column address signal, CAS, is brought low at T4, causing the memory units to latch onto this 2nd group of seven column address signals.

Now that the memory units have all 14 address signals, the particular memory location addressed will accept whatever is on the data bus until time T5, when the WR signal goes high. This forces the memory cells to latch the data and remember it, until either new data is written to that same location or the power is removed.

Control Signals

Figure 1 shows the control signals needed for proper memory access, as well as showing how they are obtained. They are as follows:

RD 32K-48K- When this signal is low it indicates a read operation from either 32 or 48K memory block. (The CAS signal, described below, determines which block). Z37 and two NAND gates (of Z20) are connected such that a low on the RD lead, and a low on either of the 32K or 48K decoded signal leads, from Figure 3, will produce a low on the output of Z20, pin 3. This RD 32-48K signal goes to Figure 2 and enables the memory output buffers, which in turn allows the memory data to be placed on the data bus.

RAS - Actually, this is a memory request signal. When low, it indicates that a memory address is being sent out on the address bus to all blocks of memory. The RAS signal is only buffered by an inverter (part of Z22) before being sent to Figure 2.

WR - This is the write lead discussed earlier in the example for Chart 2. When low, it indicates a write operation. It also is buffered by an inverter (part of Z22) before being sent to the memory IC's.

32K-CAS & 48K-CAS - These two signals are obtained from a combination of the CAS signal for the interface bus and the decoded 32K and 48K signals from Figure 3. Only when the CAS signal goes low, are the two OR gates (of Z24) in such a

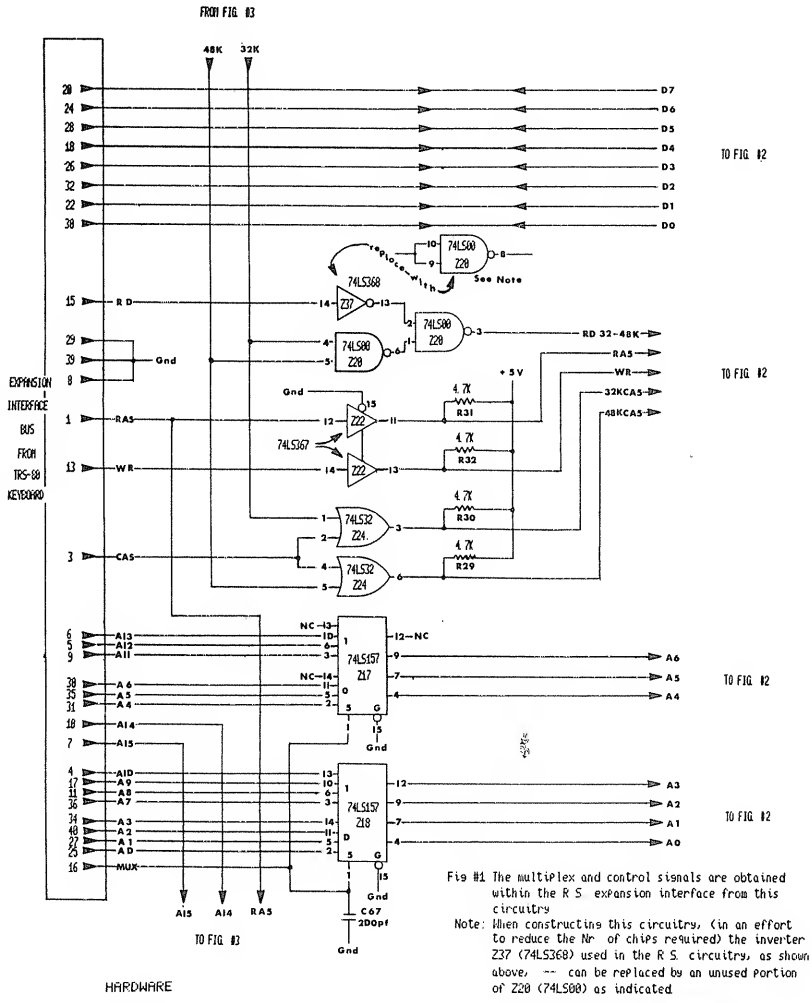


Fig #1 The multiplex and control signals are obtained within the R S expansion interface from this circuitry

Note: When constructing this circuitry, (in an effort to reduce the Nr of chips required) the inverter Z37 (74LS368) used in the R S circuitry, as shown above, -- can be replaced by an unused portion of Z20 (74LS08) as indicated

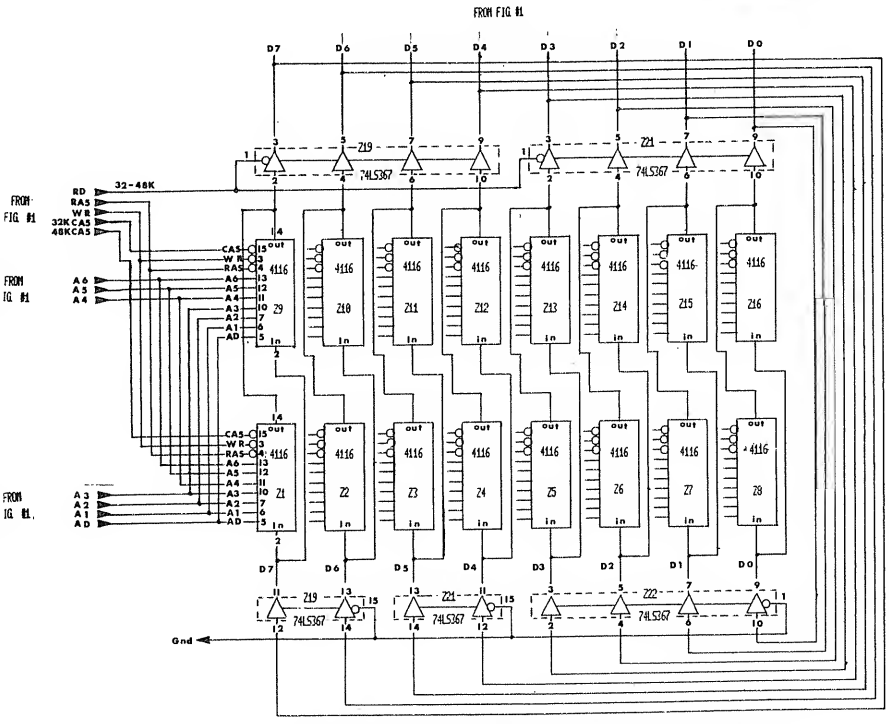


Fig # 2 The actual memory units with their input/output buffers are wired within the R S expansion interface as shown above

condition that allows a low from either of the 32K or 48K decoded signals from Figure 3 to be passed onto Figure 2 via the 32K-RAS or 48K-RAS. Since only one of these signals will be low at a time, only that particular memory block will receive the column portion of the address signals, as described earlier in Chart 2. This results in only that memory block (32K or 48K) being accessed.

Before leaving Figure 1, let's mention that resistors R29 through R32 are used to aid in suppressing unwanted negative spikes. Since, under normal conditions, these signals are held high, the resistors are connected to +5 volts. The purpose of capacitor C67 on the MUX lead is for the opposite reason. It is to suppress unwanted positive spikes. Since the MUX lead is normally held low, the capacitor is connected to ground.

Memory Block Decoding

Since only one 16K memory block can be accessed at a time, a provision for decoding the upper portion of the address is needed to determine which memory block is being requested. This decoding circuitry within the Radio Shack expansion unit is combined with the decoding of several other signals, and will be discussed in detail in a later article. Figure 3 shows a portion of the circuit which when wired alone will accomplish the same thing.

Only when the RAS signal in Figure 3 goes low, will Z43 become operational. At this time, if address leads A14 and A15 contain a low and a high respectively, then a low will be output on pin 6, which is the 32K lead. This low, when combined with a low on the CAS signal produces a low on the 32K-CAS lead, allowing only the 32K memory block to accept the total 14 address signals and become active. If A14 and A15 each contain a high, then a low will be output on pin 7, which is the 48K lead. This, when combined with the CAS signal, enables the 48K memory block.

The purpose of R35 is to suppress unwanted positive spikes on the RAS lead when it is held low. Positive spikes at this point would allow unwanted memory block to pulsed on and off.

Power Supply & Connections

The power connections for the various IC's are indicated in Figure 4. The memory IC's require three supplies, +5, -5 and +12 volts, while all other IC's require only +5 volts. The power supply circuitry used in the Radio Shack expansion interface also will be discussed in detail in a later article, but for those who wish to construct this circuit, I have included a simple configuration in Figure 4 which will do the job. It can be built for less than $9.00 if the parts are purchased from "Poly Paks" or "Hobby World".

Construction Aids

All parts including the power supply can be placed on one board. However, keep the transformer and its AC leads as removed from the other components and wiring as possible. The memory units Z1 through Z16 should be mounted in sockets. And, while it is not necessary to mount the other IC's in sockets, I prefer to do so with all my projects.

IC Z37 in Figure 1 can be replaced with an unused portion of Z20. This eliminates the need for an extra IC, namely Z37. The RD signal (pin 25) of the interface bus, will now go to pins 9 and 10 of Z20. Instead of the output of Z37 (pin 3) going to pin 2 of Z20, the output of the new portion of Z20 (pin 8) will go to Z20, pin 2.

This entire project will have to be connected to the '80's 40 pin bus on the back of the keyboard, via a 40-pin lead ribbon cable and connector. (Now available from Radio Shack stores).

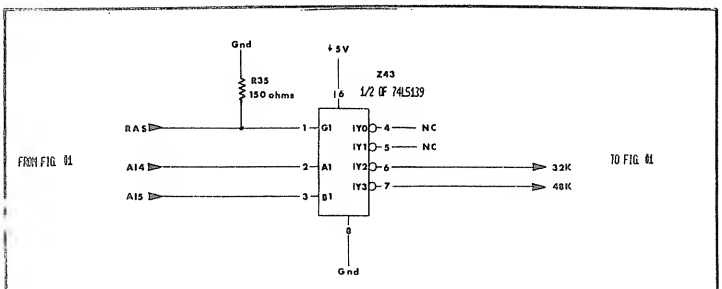
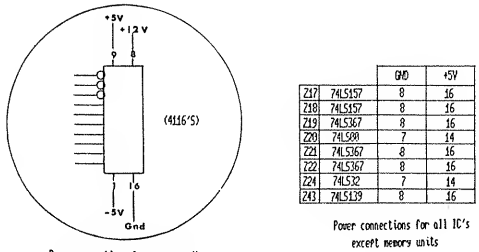


Fig #3 This decoding circuitry is needed to construct a stand-alone version of R S 32K - 48K memory expansion circuitry.

HARDWARE
MEMORY EXPANSION 32K - 48K

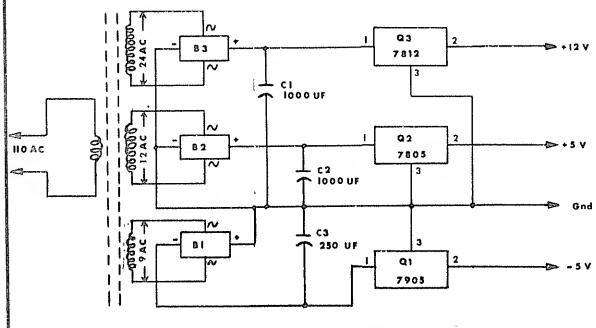


Power connections for memory units (Z1 thru Z16 (4116's)

		GND	+5V
Z17	74LS157	8	16
Z18	74LS157	8	16
Z19	74LS367	8	16
Z20	74LS00	7	14
Z21	74LS367	8	16
Z22	74LS367	8	16
Z24	74LS32	7	14
Z43	74LS139	8	16

Power connections for all IC's except memory units

B1 thru B3 = Bridge Rectifiers (50 PIV)



This power supply is capable of providing the three voltages necessary to allow the Memory Expansion Project to be self contained.

HARDWARE
MEMORY EXPANSION 32K - 48K

PARTS LIST

AMOUNT	IDENTIFICATION	DESCRIPTION
16	Z1 thru Z16	16-pin IC Sockets
2	Z17 & Z18 74LS157	Quad 2 to 1 multiplexers
3	Z19,Z21,Z22 74LS367	Tri-state Hex buffers
1	Z20 74LS00	Quad 2-input NAND gates
1	Z24 74LS32	Quad 2-input OR gates
1	Z43 74LS139	Dual 2-input decoders
3	B1,B3,B4 50 PIV	Bridge Rectifiers
1	Q1 7905	Minus 5 volt regulator
1	Q2 7805	Postive 5 volt regulator
1	Q3 7812	Positive 12 volt regulator
1	Transformer	Poly Pak #6407 $2.98
4	R29 thru R32	Resistor 4.7K ohms
1	R35	Resistor 150 ohms
1	C67	Capacitor 200 pf
2	C1 & C2	Capacitor 1000 Mf
1	C3	Capacitor 200 Mf

Add POWER to your TRS-80™

SOFTWARE by MiProg

XEDIT, a high powered compact disk based editor designed for the TRS-80™ Model I or II. Whether it is BASIC, ASSEMBLY, or FORTRAN, XEDIT is packed full of commands needed by programmers who are serious about their work. Here are just a few features:

- Edits most file formats
- Block text copy command
- Locate, Delete, and Change with windows
- Inserts and maps up to five input files
- Upper/lower case compatible
- Operates with or without line numbers
- Rapid access disk cache
- Recovers from most DOS errors
- Fast file entry point map
- Change text command for any number of occurrences
- DOS Directory and Kill commands
- Line printer paging with adjustable forms
- Sophisticated reprinting line editor, handles line feeds
- Disk BASIC, Disk EDTASM, and EDIT-80 format compatible
- Display status command, includes free memory, current pointer printer forms, number of input files, output filename and format.

XEDIT will handle files of any size up to 2.7 Megabytes or 10K lines in length. Comes complete with instructions covering operation, externals, and file formats.

Model I (32K single disk system)
- Formatted diskette $44.95
- Cassette tape. $39.95

Model II
- Formatted diskette $89.95

Model III (32K single disk system)
- Formatted diskette $79.95
- Cassette tape. $75.95

ASM/CMD, a disk based assembler which generates object code to disk or tape (disk only on Model II). Accepts any file format including ASCII Disk BASIC. Listing may be outputted to display, disk file, or paged with adjustable forms to printer. Operates under standard Z80 Zilog Mnemonics with 9 pseudo operations. Comes complete with operating manual.

Model I (16K single disk system)
- Formatted diskette $34.95
- Cassette tape. $29.95

Model II
- Formatted diskette $59.95

Model III (32K single disk system)
- Formatted diskette $49.95
- Cassette tape. $45.95

PACK/CMD removes spaces from text files generated by XEDIT, and EDIT-80 to reduce file lengths by 5 to 40 percent. PACK will also strip comment fields and line numbers for additional space savings. Text can be masked for upper case only. Does not destroy compatibility of assembly and FORTRAN source files. Comes complete with instructions.

Model I (16K single disk system)
- Formatted diskette $14.95
- Cassette tape. . . . . $ 9.95

Model II
- Formatted diskette $19.95

Model III (32K single disk system)
- Formatted diskette $14.95
- Cassette tape. . . . . $ 9.95

Special package, XEDIT, ASM, and PACK.

Model I
- Formatted diskette $79.95
- Cassette tape. $74.95

Model II
- Formatted diskette $149.95

Model III
- Formatted diskette $129.95
- Cassette tape. $125.95

XDIR/CMD, an extended directory that offers more than the standard TRSDOS™ directory. XDIR will do multiple drive directories with all file attributes including extent locations, file length, EOF index, EOF record, protection level, LRL, password indication, track lockout indication, and much more. XDIR will also display to the printer.

Model I (16K disk system)
- Formatted diskette $19.95
- Cassette tape. $15.95

CALL/CMD extends and improves the TRSDOS™ AUTO function. Can be enabled and disabled by prompts, and through keyboard, resident program, or the call file.

Model I (16K single disk system)
- Formatted diskette $19.95
- Cassette tape. $15.95

TANDON/CMD improves TRSDOS™ by allowing higher step rate, extending access to 40 tracks for the new Tandon disk drives. Also fixes the break key problem.

Model I (16K single disk system)
- Formatted diskette $14.95
- Cassette tape. . . . . $ 9.95

DEXER/CMD, a disk exerciser emulator program designed to speed repair of any TRS-80™ compatible disk drive. DEXER eliminates the need for the Shugart SA809 test fixture and decreases repair time with easy to use commands and on screen display of required set up data. DEXER was written specifically for the repair technician and Shugart or Tandon disk drives. Shugart alignment diskette or equivalent and a 30Mhz oscilloscope required. One key commands allow easier adjustments necessary for Shugart alignment. DEXER is not for general disk testing and is recommended only for service personnel who have previous experience in disk drive repair.

Model I (16K single disk system)
- Formatted diskette $24.95
- Cassette tape. $19.95

Dip shunts for conversion and upgrades for the TRS-80™. Comes complete with instructions for A, D, E, and G level boards and new 2 chip level II.

Two dip shunts and instructions $1.00

Please send check or money order to:

MiProg
P. O. Box 27014
Minneapolis, MN
55427
612-574-1711

Minnesota residents add 4% sales tax. Outside continential U.S.A., add $3.50 postage and handling.

TRS-80™ and TRSDOS™ is a trademark of Radio Shack, a Division of Tandy Corporation.

Six Ways To Get More Out Of *SCRIPSIT*

Some ways to make Radio Shack's Scripsit and Apparat's NEWDOS/80 work better for you.

David D Busch
Revenna, Ohio

Very powerful programs are seldom used to their fullest capacity, and Radio Shack's Scripsit is no exception. One of the most fascinating aspects of learning to use a good word processing program like Scripsit is the never ending discovery of ways time and effort can be saved by applying features in new ways.

Here are six ways of using Scripsit that you may -- or may not -- have thought of. Some may be equally applicable to other word processing programs for the TRS-80, such as Electric Pencil.

1. Yes, you can get a directory of files available without exiting Scripsit. Enterprising machine-language programmers already have patches on the market that will accomplish this, but NEWDOS 80 owners don't even have to go to the extra expense or bother. The 'DFG' invocation of Mini-Dos works from Scripsit.

Merely hit the three keys together, and type DIR when the Mini-Dos prompt appears. Then type MDRET to return to the normal Scripsit mode. Your text will be gone from the screen, but it may be retrieved just by hitting BREAK and ENTER consecutively. This returns an INVALID COMMAND message below the status line, but does no harm.

You probably will have an unwanted character on the screen (a lowercase d, f, or g), but backspacing and making a one-character correction is a small price to pay for a major patch to an otherwise fine program.

NEWDOS 80 will also allow you, through the SYSTEM command, to set up new default drive numbers for DIR or program SAVES. Multiple drive owners can keep a system disk in Drive 0, and save all Scripsit files on data diskettes located in another drive, without having to append a drive specification onto the file name.

2. Eliminate paper entirely. Scripsit's low cost, compared to most other word processing software, is making it one of the most common programs among TRS-80 owners. In my users' group, nearly every member uses Scripsit, as do most of my TRS-80 computer-using friends. It is not unreasonable to assume that most microcomputer magazines also have their own copies running for evaluation.

In such cases, why bother to distribute paper printouts of Scripsit text files at all? Correspondence between Scripsit-owning friends might just as well be in disk form as paper. One disk exchanged between pen-pals might very well be used and reused dozens of times before it was damaged or worn out. As a bonus, it's a simple matter to keep a number of letters on one disk. Old correspondence can remain available for reference for a long time, until the most aged letter is finally erased (or copied onto another disk) to make room for a new missive.

Articles submitted to microcomputer magazines with compatible programming don't have to be retyped after the editor has looked the material over. Editing can be done at the computer, and then the file printed out for re-entry into the publication's phototypsetting equipment. Given a large number of Scripsit or Electric Pencil disk submissions, it's not unreasonable to assume that someone would be motivated to construct hardware and software interfacing between the TRS-80 and phototypsetting gear.

Some users groups are already taking advantage of modem-linked microcomputers to compose their newsletters as a group effort. Various contributors relay submissions by phone, and then the newsletter is printed out in one central location.

Why print out the newsletter at all? The computer lends itself to the creation of a really dynamic document, a publication that has no deadlines, no closing dates, and is never outdated. Members can add articles as they wish, scan the current table of contents to see what other articles are available, and retrieve and read them at their leisure.

Instead of waiting a month for the next issue, members could check out their club newsletter as often -- or as seldom -- as they wished. After a

month or two, an article could be dropped to make room for new contributions.

Those members of a club without modems could still access the newsletter. At each club meeting, several copies of the newsletter, on disk, would be made available for duplication. Then, readers could take the disk home and pursue it at their leisure by scanning the Scripsit file.

This scenario assumes, of course, that everyone has a copy of Scripsit. That's not essential, however. Because Scripsit files are saved in ASCII form, it's simple to write a BASIC program that will read a Scripsit file and display it on the screen (interpreting the various control codes is a different matter, however).

Non-disk owners in a club could be accommmodated by printing a copy or two of the newsletter at intervals.

3. Don't type out complicated names. In my work as a writer, I frequently do articles about people or companies with long, complicated, or hard to spell names. No matter what your application for Scripsit is, you will encounter equipment or brand names or other bothersome terms that will be used repeatedly in a document. I didn't bother to enter the word Scripsit each time I used it in this article.

Instead, I frequently take advantage of the not always understood Global Search and Replace command. When entering text, use some convenient abbreviation for any words that you don't want to spell out. For Scripsit, I used the lowercase letter "s". Interglobal Amalgamated Brotherhood of Pipefitters could be shortened to "i".

The abbreviation should be some character or characters that will not appear in that form within the text, except in places where you want it to stand for the other word or words.

To perform the Global Search and Replace, first hit control "R" (@ R) to invoke the repeat command function. Scripsit will ask you how many times you want the command repeated. Enter 255 (the maximum) to be on the safe side, then hit ENTER.

The next prompt you see below the status line asks you to enter the command you want repeated. Hit BREAK, then type an R, the "greater than" sign, the string you want replaced, another "greater than" sign, followed by the string you want substituted.

It's important to type both *exactly* as they appear -- including blanks. To substitute the word "Scripsit" for the letter "s", I didn't really want every letter "s" to be replaced, just those surrounded by blanks. So, I entered:

R› s › Scripsit

That's R› (space)s(space)› (space) Scripsit. Only solitary "s"'s in the text were searched for and replaced.

By the way, the Repeat command is also not used as often as it could be. Need three copies of a document? Hit control R, the numeral 3, then BREAK, and a P.

4. Scripsit makes it easy to jump to the beginning or end of a document by hitting shift up and down arrows. But how do you get to a specific point in the middle of a large document without scrolling endlessly?

Global Find will look for you. There are two ways of searching. One is to look for a specific string. You must recall a phrase used in the part of the document you want to access. The phrase might be a subhead or chapter heading, or the name of someone mentioned in the section of text.

Type BREAK, F› "text string looked for". Scripsit will stop at the first occurence of the string you have input,then retain the string in memory. Wrong place? Just hit BREAK, FX, and the cursor will move to the next place the string appears. It is not necessary to re-enter the string.

You can also search for specific blocks. As text is entered, deposit block markers at strategic points, with a control Q, and "label" (A,B,C, etc.). End the block with a control Q followed by a control down arrow. Text can also be marked into blocks in this manner after it has been entered, if desired.

To find a specific block, jump to the beginning of the text, and enter the Global Find command exactly as before, except instead of entering a string, enter the name of the block exactly as you did when the block was created. Hit a control Q, and the label of the block. The block bracket figure will appear on the command line, and when you hit enter, the cursor will jump to that point in the text.

5. Use standard headings and other blocks of text over and over. Certain types of articles that I prepare have components that are used nearly every single time I prepare that type of document. I have a standard Scripsit file called ARTICLES, that I load each time I begin a new project. It begins with the standard header-footer format that I use, it has a cover sheet, photo caption sheet, form letter to my client, and a photo caption page already laid out with the standard information.

Within each of these are repeated uses of words such as CLIENT, TITLE, etc. By using Global Search and Replace, I can substitute the name of the current client and project in a dozen places with a few keystrokes. Then specific information, such as picture captions and price for the project, can be filled in and the main text begun. Twenty minutes of detail work is finished in three.

Perhaps you have a similar "boilerplate" that can be reused. It need not appear at the beginning of a document. By preparing the material in advance, giving each an appropriate file name, and then using the load and chain command, the standard material can be inserted where desired. (The correct procedure is: BREAK, L, (comma),C, (comma), (file name).

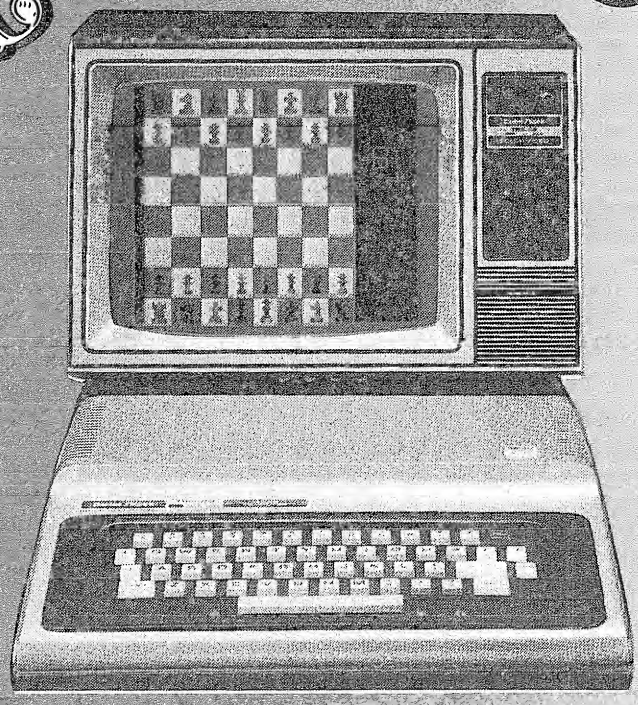
6. Does your company have a document that must be frequently updated? Don't print even one master copy. Store the text as a Scripsit file instead, make updates as needed, and print out a copy only at the appropriate time.

My firm, for example, has an employee policy and benefits guidebook, which is very finely detailed -- only because we can update it anytime we wish. We may change the mileage or per diem allowance every year, raise prices, add or drop clients, and change benefits. But, each time a new employee joins my company, he or she receives a freshly printed out copy of the very latest information. Memos keep everyone else apprised of changes when they occur, and once a year, everyone receives a new policy book.

Conclusion

Scripsit is a highly-flexible word processing program that can do a great deal more than even veteran users realize. We are all buried in preconceptions that have built up through years of using paper and doing things manually. Sometimes it takes a lot of re-thinking to realize that, with a computer, sometimes paper isn't even necessary for a given application. Or that something we have always done by hand can much more quickly be performed by the computer if we only had the sense to ask it.

The Color Computer



Will it Replace the *CRAY 1?*

William Barden, Jr.

At the last meeting of our TRS-80 User's Group, I took an impromptu survey. I struggled to my feet amidst a barrage of discarded cassette tapes, burnt-out RAM chips, and marked-up copies of "TRS-80 Assembly-Language Programming."

"How many of you are interested in the Color Computer?", I shouted. Immediately, hands shot up all over the place.

"Looks like three," I muttered. It turned out later, it was three *hands*... Well, one and one-half interested users aren't bad...

Now it's obvious from first reports that the Color Computer is going to sell very well among first time computer users. But what about sales to Model I and III owners? The purpose of this article is to tell you, as a suave, sophisticated user of the Model I or III, why you *should* be interested in the Color Computer - and not just because I'm a paid shill of Radio Shack either. (I don't need Radio Shack - I could always go back to my TRSDOS diskette bulk erasing business...)

What's Inside the Color Computer?

The Color Computer comes attractively packaged in standard Radio Shack battleship gray. Physically, it's about the size of the Apple - small enough to put in a large briefcase, for those trips to Fort Worth. Of course, the $399 price does not include a television, but the built-in RF converter allows you to hook it up to a color TV at the Fort Worth Hilton.

Before the hard sell, let's take a look at the design of the Color Computer. The basic pieces are shown in Figure 1. They are CPU, video display logic, RAM, ROM cartridge, sound generator, joystick inputs, and serial port.

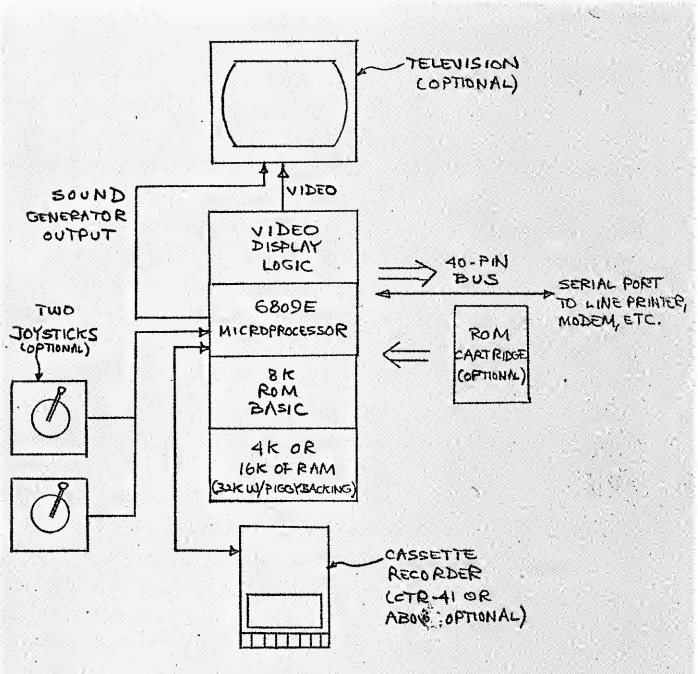


Figure 1. Color Computer Functional Blocks

The 12-Bit 6809E CPU

Stop, stop! - I know that it's not truly 12 bits! The Motorola 6809E microprocessor is a transition mircroprocessor - it's basically an 8-bit microprocessor with 16-bit addressing capability. The 6809E is an upgrade of the Motorola 6800 microprocessor, which was (and is) a nicely designed microprocessor with an easy to use instruction set. Of the Z-80, 8080 (S-100 microcomputers), 6800 (Southwest Technical Products and others), 6502 (Apple, KIM, SYM, AIM, PET), I like the 6800 the best as far as instruction set. However, the microprocessor is such a small portion of the total package....

The 6809E includes the 6800 instructions as a subset. The 6800 instructions are compatible on an assembly-language, and not machine-language level. It means that the machine code for 6800 programs would have to be reassembled because of minor differences in OP code formats for the 6809E.

In addition to the 6800 subset, the 6809E extends the basic instruction set by more powerful branch, indexing, and stack operations. The biggest marketing blurb, of course, is the hardware multiply. Hardware divide is not present.

The 6809E has a nice instruction set, and if you like assembly-language programming, you'll enjoy working with it. Radio Shack has definite plans for assembly-language on the Color Computer, and there are already several assemblers available for the 6809E from other sources.

High Resolution Color Graphics

The most noticeable feature about the Color Computer (unless you are operating sans color TV) is the Color Graphics. The design uses the Motorola 6847 Video Display Generator chip. One gets the feeling that the rest of the computer was designed around this chip.

The VDG chip is an LSI color graphics chip that enables a number of different modes. Basically there are three modes - alphanumeric, "semi" graphics, and full graphics.

The alphanumeric mode gives you color alphanumeric characters arranged in 16 lines of 32 characters. I can hear some of you groaning out there. A line of 32 characters is rather short, but suffices for many BASIC and data entry lines - you're just going to have to accept it!

The semigraphics mode is a VDG mode that allows up to 64 horizontal elements by 192 vertical elements. Not all elements are separetely addressable. The full graphics mode allows up to 256 elements horizontal by 192 elements vertical, each separately addressable. Generally, the higher the resolution, the more limited the number of colors; the highest resolution mode (256 by 192) allows two colors, while the next highest (128 by 192) allows four colors.

I can see Apple and S-100 microcomputer owners sneering at the Color Computer's graphics resolution. There's no question that it's nice to have as much resolution as possible for some graphics applications, but even a 128 by 192 in four colors is 4 times the resolution of the Model I, and should be fine for most applications. Since the memory mapping is much more straightforward in graphics mode than the Model I (no more six bits in one byte), the graphics are easier to program and can be thrown up on the screen at a much faster rate.

"I can see Apple and S-100 computer owners sneering at the Color Computer's graphics resolution."

Memory Mapping

The memory map of the Color Computer is shown in figure 2. RAM occupies the first 4K of 16K memory. The screen is memory mapped into the RAM area. At a minimum the screen takes up 32 characters per line by 16 lines, or 512 characters. When the highest resolution "semi" graphics or full graphics modes are used, the screen area occupies 6,144 bytes (256*192/8). One strongly suspects that an upgrade from 4K RAM to 16K RAM would be almost a necessity to do any high-resolution graphics work. The high-resolution modes would leave 10K bytes available for user program area.

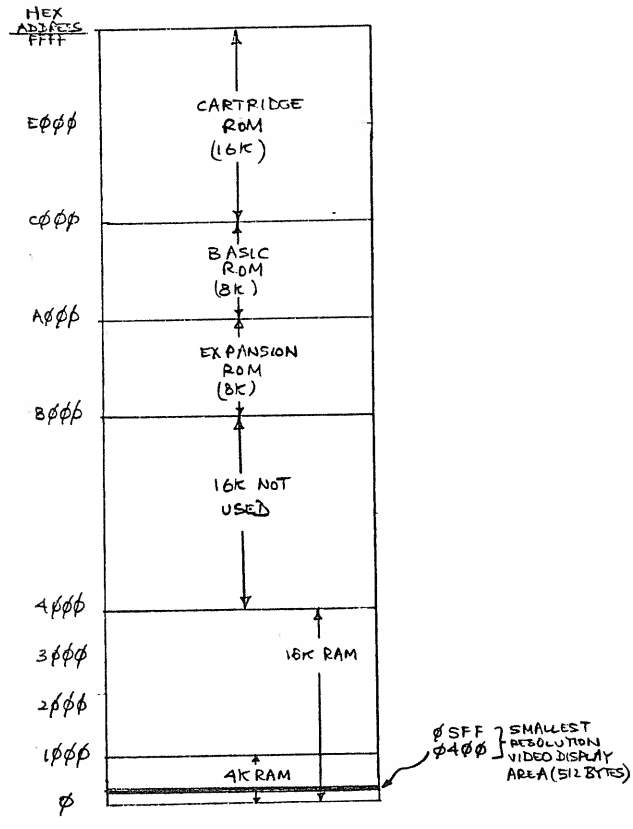


Figure 2. Color Computer Memory Map

RAM used in the Color Computer is our old friend, the 4116 dynamic RAM. True to form, users are already buying RAM and installing it themselves.

How about the 16K "not used" area? Under certain conditions, it appears that an additional 16K of RAM could be "piggybacked" to provide 32K bytes of RAM.

The basic "Color BASIC" is 8K of ROM; the expansion ROM occupies another 8K bytes. Note that the mapping scheme is inverted from the Model I/III.

The upper 16K memory is dedicated to "cartridge" ROM. Cartridge ROM is exactly that - a plug in cartridge with 2716-type EPROM that plugs into a 40-pin edge connector. You may have seen the piles of cartridge games at your local Radio Shack - there will be a proliferation of such software for the cartridge area. The interesting thing about the 40-pin edge connector, is that it brings out all system signals! Radio Shack makes no bones about the fact that any type of expansion should be possible via this connector, and I can think of some great applications!

Six Bits for A D-to-A?

One of the most interesting remaining features on the Color Computer is a six bit "digital-to-analog converter." Sound from the Color Computer is produced either by a one bit output, or by the six bit output. Both feed into the audio channel of the modulated RF signal to the television. (Cassette tape audio can also be fed into the TV audio, a handy feature for monitoring cassette program input, or for audio sound effects or narration in programs!).

The six bit D-to-A is used to generate the cassette output for the Color Computer, which is "frequency shift keying" of 1200 or 2400 hertz at a speed of 1500 baud (three times the baud rate of the Model 1).

The D-to-A can be programmed via assembly language or BASIC commands to generate music or other sounds. The Extended BASIC, for example, has a "PLAY string" command which lets you set the volume, note, and tempo of a string of notes.

Virtually any sound can be created by the D-to-A scheme, as it allows synthesis of 64 different output levels, allowing for generation of any waveshape from violins to Texas accents (shudder). (Of course, as with the graphics, there is a slight problem - generating and retrieving the data base to produce the sounds - but the capability is there).

Analog Inputs

The inverse of the D-to-A is the A-to-D capability. This is implemented by two joystick inputs. A friend of mine expressed interest in the Color Computer primarily because of the set of inexpensive joysticks which he wanted to use in other projects!

The joysticks, of course, are really just potentiometers for the X and Y direction. A BASIC command (JOYSTK) allows you to read either joystick and the "vertical" or "horizontal" position of each. Of course, assembly language would also allow you to read the four inputs. Substitute other analog inputs for the four channels and you have spiffy A-to-D inputs which can be used for reading temperature, windspeed, voice level, or any other physical signal that can be converted to a voltage level!

Serial Devices

One of the nicest aspects of the Color Computer is that you don't have to keep reaching into your wallet to pay for new options. The serial channel is already there! The Color Computer does everything in software, where possible. This isn't necessarily bad. The software drivers for serial input and output are already in the Color Computer. The RS-232-C outputs are via DIN connectors on the back of the unit.

The signals available on the DIN connector are RS232IN, RS232OUT, GROUND, and CD for status. The baud rate for data transmission is software selectable from 120 to 2400 baud. Printer parameters such as line width can also be set in the software.

Alas, there is no Centronics parallel port. How do you then drive your existing Line Printer I, II, III, IV, or XVIII with the Color Computer? One alternative is to

add a serial to parallel converter, opposite in function to Small Systems Software TRS-232. A second method is to go into an existing Model I/III via the RS-232 and then out to the printer. Some printers, of course, have both serial and parallel capability.

Other Goodies

Also presumably in the works are disk drives for the system. Whether these will be single or double density is anybody's guess.

WHAT'S OUTSIDE THE COLOR COMPUTER?

The Keyboard

Figure 3 shows the keyboard layout. Right away I can hear snide comparisons to the first PET keyboard and the TI-99 keyboard. I have used the keyboard, and I can testify that while it is not equivalent to an IBM Model 60 typewriter, it is not that bad, either. I would think that a computer of this type would not make a good word processor primarily because of the large character format color display, rather than because of the keyboard action. There...ahem...appears to be no keyboard bounce.

Interfacing

The back panel of the Color Computer is shown in figure 4. There is a RESET and POWER switch and a switch to select output on either channel 3 or 4. Four DIN connectors provide interface connections. Two of the connectors are for the joysticks, one is for the RS-232-C input/output, and the fourth is a cassette tape jack.

The cassette tape connector, by the way is not included with the system (nor is the cassette recorder). When I tried to put in my Model I cassette plug, it was no go. A quick trip down to the local Radio Shack yielded a DIN plug with a thin metal wall. Several hours later, I still had not successfully loaded tape. I finally isolated the problem down to one unit - the television! Sure enough, when the TV was turned off the cassette tapes would load fine over a fairly wide range of volume settings. Serves me right for getting a cheap TV instead of an approved RS model! The TV is now physically located one room away from the system, and I've added a pair of binoculars to the setup (but seriously...).

Documentation

Documentation for the Color Computer consists of a 32-page operation manual, a nicely presented reference card, and a David Lien-type book called "Getting Started with Color BASIC." Sure enough, the latter features a cousin of the Model I dancing computer and is geared to the beginning user with no prior computer experience. The BASIC book is excellent for this user, but the experienced Model I user will probably go right to the reference card.

Color BASIC and Extended BASIC

The basic BASIC, which RS calls "Color BASIC", is somewhere between Level I (remember that?) and Level II in capability. String functions such as MID$, LEN, LEFT$ and RIGHT$ are implemented. One-dimensional arrays in 4K RAM and multi-

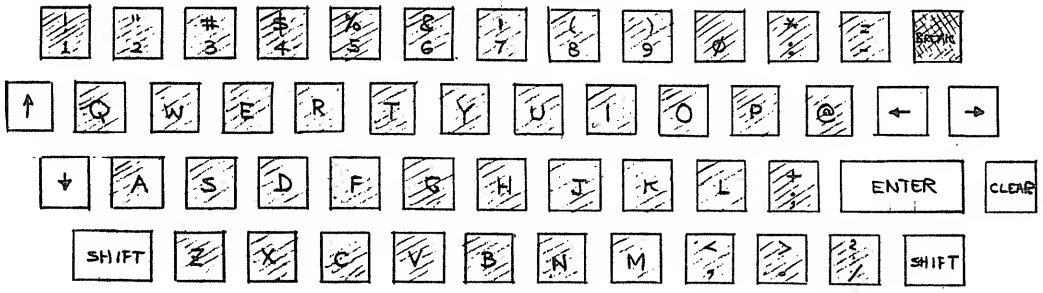


FIGURE 3

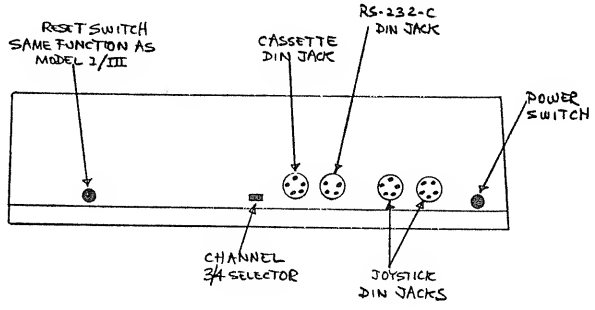


Figure 4. Color Computer back panel.

dimensional arrays above 4K are permitted.

Table 1 shows the BASIC commands and functions that are identical, or very similar, to Level I/III BASIC. Some additional commands unique to the Color Computer are:

1. SET/RESET/POINT These are similar to the graphics commands for the Model I/III but allow a 64 by 32 resolution in any of 8 colors.

2. CLS clears the screen to any one of 8 colors.

3. OPEN and CLOSE opens or closes a file at screen, keyboard, cassette, or line printer. EOF tests whether an end-of-file has been reached. Interesting commands - they promise some device independent I/O (?).

4. Cassette commands other than CLOAD and CSAVE include CLOADM, which loads a machine-language tape, MOTOR, which turns the cassette motor on or off, and SKIPF, which finds a specified cassette file. The AUDIO command connects the cassette output to the TV sound channel.

5. EXEC is similar to the transfer command of SYSTEM. It transfers control to a machine-language program.

6. PRINT#-2 is identical to LPRINT, printing a list of items on the system line printer.

7. SOUND outputs a tone of a frequency of 1 to 255 and duration 1 to 255.

8. JOYSTK returns the horizontal or vertical position of joystick 0 or 1.

Table 1
Color Basic Commands and Functions Identical

To Level I/III Basic

ABS	MID$
ASC	NEW
CHR$	ON...GOSUB
CLEAR	ON...GOTO
CLOAD	PEEK
CONT	POKE
CSAVE	PRINT
DATA	PRINT@
DIM	PRINT#-1
END	PRINT TAB
FOR...TO...STEP/NEXT	READ
GOSUB	REM
GOTO	RESTORE
IF...THEN...ELSE	RETURN
INKEY$	RIGHT$
INPUT	RND
INPUT#-1	RUN
INT	SGN
LEFT$	SIN
LEN	STOP
LIST	STR$
LLIST	USR
MEM	VAL

The Extended BASIC should be out by the appearance of this article. It includes a full set of graphics commands which operate at high speed, judging from what I saw in Fort Worth. Commands are included to draw a line, rectangle, circle, ellipse, or arc, in addition to such things as setting the screen colors. Other commands upgrade the BASIC to something approaching Level II, including such assembly-language hooks as VARPTR and USRn. Another thing that the Extended BASIC provided is full line editing, which is quite conspicuous by its absence in Color BASIC. (Whhy are all typographical errors in the first two charcters of a line?)

Why I Think You Should Buy a Color Computer

It's very difficult to convince microcomputer users about the merits of other systems. A strain of cussedness runs through the fabric of all microcomputer users; they tend to pan the competing systems and be very chauvinistic about what they have, even if it is an Altair 8800. However, try to take an unbiased look at the Color Computer. I would say that it might be particularly interesting in the following areas:

1. Experimenters. Here's a state of the art microprocessor with hardware multiply, D-to-A and A-to-D channels, and a 40-pin bus hanging there ready for all kinds of devices. How about EPROM programmers, dedicated systems for home control, ham radio applications (good shielding, here), and others?

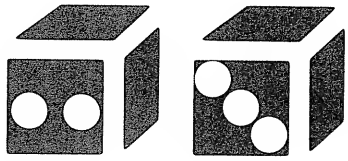
2. Software freaks. Here again, the 6809E is perfect for an assembly-language programmer. There are some tools in the way of disassemblers, debuggers, and assemblers already, and there'll be a lot more. The 6809E is faster than the Model I Z-80 and will lend itself to all kinds of applications programs.

3. Graphics work. Maximum resolution is 256 by 256, which isn't bad. Since the memory mapping of the Color Computer is a lot more straightforward than the Model I, graphics work will be a lot easier. Three dimensional display work is not the easiest thing in the world, but this machine should make it much more palatable than the Model I.

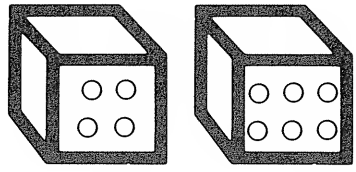
4. Intelligent peripherals. Need an intelligent peripheral for your existing Model I or II? For $400, you can't beat the versatility of a complete computer system! Use *it* as a color graphics controller!...Or let it sit there with the modem collecting data while you use your Model I or III for more important processing.

5. Music buffs. The six-bit D-to-A offers pretty much unlimited capability for sound synthesis. Coupled with the 6809E instruction set and hardware multiply, you've got a powerful system for this application.

6. Writers. Get a Color Computer, research it, and write a number of articles for computing magazines. I can verify that area is especially interesting... ●



Computer Yahtzee



This program works without modification on Models I and III, 16K Level II and up.

Yahtzee is a Registered Trademark of L S Lowe & Co, a Division of Milton-Bradley Inc.

John A Jagos
For Lowe, Weeks Co

Computer Yahtzee is based on the game of Yahtzee that has been played by people for generations. It may only be played by two players at one time, and you can only have one Yahtzee per game. Your dice are numbered one to five. To re-roll the desired dice you simply state the number or numbers of those dice. If you do not wish to re-roll you type in a zero. The score card is numbered one to thirteen in the left column. At this time, you simply state which box you wish to score in by typing in that number. You then press ENTER and the game continues.

In this game the player is allowed up to three rolls of the dice, in order to get a scoring combination for each round. Although a player is allowed up to three rolls, he may stop after the first or second roll if he wishes. After the first roll, the player may re-roll any or all of the dice. He does not have to state what he is going for and he may change his mind at any time. After the third roll the dice are final and *must be scored.*

Each player has his own scoring column. He has to make up his own mind of what and where he wishes to score, according to his own judgement. The computer will score you according to your choice. The game ends after all the boxes have been filled by you or your opponent. The player with the highest score wins.

The thirteen scoring boxes consist of Aces, Twos, Threes, Fours, Fives, and Sixes, which make up the Upper Section, and three of a kind, four of a kind, full house, small straight, large straight, Yahtzee and chance, which make up the Lower Section.

Use Strategy

If all appropriate boxes are already filled, you would have to score a -1 in another box. It would be best if you could take a -1 in an open box of the Upper Section without ruining your chances for making 63, and earn the 35 bonus points. If this cannot be done, a player would have to use his own judgement in placing the -1, so as to lose the minimum number of points.

The Upper Section

If a player chooses to score in the Upper Section he should know that only dice with that number on them count toward his score. For example, if you should have three dice with fours on them and one die with a six and one with a three, when you put it in your fours box you would get the socre of twelve. If a player scores sixty-three or more in the Upper Section he earns a bonus score of thirty-five points.

The Lower Section

Each category of the Lower Section explains itself. Yahtzee is when all five of the dice show the same value, and it scores 50 points. The chance box may be filled at any time. If you should have no other combination, this is a great time to take advantage of chance. Your score for chance is the total of all dice. ●

```
0 CLS:PRINT"Y A H T Z E E":PRINT"BY JOHN A JUNOD  DEC 80"
0 PRINT:PRINT"  THIS IS A TWO PLAYER GAME  AND FOLLOWS  THE RU
    LES FOR":PRINT"NORMAL  YAHTZEE.  ONLY ONE YAHTZEE IS ALLOWE
    D. (SORRY)":PRINT"IF YOU HAVE A SECOND ONE , YOU WILL HAVE
    TO USE IT ANY-":PRINT"WHERE YOU CAN.":PRINT
0 PRINT"  YOU MAY PRESS ( L ) WHEN THE DICE ARE DISPLAYED TO '
    LOOK' AT THE SCORE CARD WITHOUT DISTURBING YOUR ROLL. IF YO
    U ATTEMPT TO PLACE A ROLL WHERE IT CANNOT GO, YOU WILL RECI
    EVE A -1 IN THAT  POSITION.":PRINT
0 CLEAR1000:DIMA(13):DIMB(13):DIMC(13):DIMD(5):DIMA$(13):G$="#
    #":PP=1
0 FORZ=1TO13:READA$:A$(Z)=A$:NEXT
```

```
50 INPUT"PLAYER #1 ";N$(1):INPUT"PLAYER #2 ";N$(2)
70 DATA"ONES","TWOS","THREES","FOURS","FIVES","SIXES"," 3 OF A
   KIND"," 4 OF A KIND"," FULL HOUSE","SM STRAIGHT","LG STRAIGH
   T","YAHTZEE","CHANCE"
80 FORZ=1TO13:READA:C(Z)=A:NEXT
90 DATA3,6,9,12,15,18,0,0,25,30,45,50,0
100 DATA449,454,459,464,469
110 FORZ=1TO5:READA:D(Z)=A:NEXT
120 GOTO570
130 REM                    ((((( DISPLAY SCORE CARD )))))
140 CLS:PRINT"       Y A H T Z E E";TAB(29)N$(1);TAB(39)N$(2)
150 FORZ=1TO6:PRINTZ"  "A$(Z);TAB(18)USINGQ$;C(Z);:PRINTTAB(30)
   USINGQ$;A(Z);:PRINTTAB(40)USINGQ$;B(Z);:PRINT"  "CHR$(191):N
   EXT
160 PRINT"TOTAL (+BONUS)    63 (+ 35)";TAB(29)USING"###";T1(1);
   :PRINT" ======";TAB(39)USING"###";T1(2);:PRINT"  "CHR$(191)
170 FORZ=7TO13:PRINTZ" "A$(Z);TAB(18)USINGQ$;C(Z);:PRINTTAB(30)
   USINGQ$;A(Z);:PRINTTAB(40)USINGQ$;B(Z);:PRINT"  "CHR$(191):N
   EXT
180 PRINT"TOTAL  TOP & BOTTOM";TAB(29)USING"###";T2(1);:PRINT"
   ======";TAB(39)USING"###";T2(2);:PRINT"  "CHR$(191);
190 IFXX=1PRINT@240,"PRESS ANY KEY..";
200 IFXX=1THENIFINKEY$=""THEN200ELSERETURN
210 IFZZ=1PRINT@174,N$(WW)" WON!!";:PRINT@1010,"";:XX=1:GOSUB1
   90:RUN
220 REM                    (((( ENTER POSITION )))
230 L=174:PRINT@L,N$(PP)"'S ROLL WAS :";:PRINT@L+64,R(1)R(2)R(3
   )R(4)R(5);
240 PRINT@L+192,"WHERE DO YOU WANT";:PRINT@L+256,"THE SCORE?";
250 K$=INKEY$:IFK$=""THEN250ELSEF=ASC(K$):IFF(49ORF)57ANDF()78T
   HEN250
260 PRINT" "K$;:IFF()49THENG=9:GOTO290
270 K$=INKEY$:IFPEEK(14400)=1THENG=9:GOTO300:ELSEIFK$=""THEN270
   ELSEG=ASC(K$):IFG(48ORG)51THEN270
280 PRINTK$;
290 IFPEEK(14400)()1THEN290
300 PL=F-48:IFG()9THENPL=(F-48)*10+G-48
310 IFPP=1ANDA(PL)()0PRINT@L+263,"    ";:GOTO240
320 IFPP=2ANDB(PL)()0PRINT@L+263,"    ";:GOTO240
330 REM                    ((( FIND CHEATERS )))
340 X=0:IFPL(7THENFORZ=1TO5:IFR(Z)=PLTHENX=X+PL:NEXT:ELSENEXT
350 IFPL(7THENIFX=0THENX=-1:GOTO510:ELSEGOTO510
360 IFPL=7ORPL=8ORPL=13THENX=0:FORZ=1TO5:X=X+R(Z):NEXT:IFPL=13T
   HEN510
370 IFPL()7THEN390ELSEC1=0:C2=0:FORZ=1TO5:C1=R(Z):C2=0:FORY=1TO
   5:IFC1=R(Y)THENC2=C2+1
380 IFC2)=3THEN510ELSENEXTY:NEXTZ:GOTO480
390 IFPL()8THEN410ELSEFORZ=1TO5:C1=R(Z):C2=0:FORY=1TO5:IFC1=R(Y
   )THENC2=C2+1
400 IFC2)=4THEN510ELSENEXTY:NEXTZ:GOTO480
410 IFPL()9THEN430ELSEFORZ=1TO5:C3=0:C1=R(Z):C2=0:C3=0:FORY=1TO
   5:IFC1=R(Y)THENC2=C2+1ELSEC3=C3+R(Y):IFR(Y)=C3/2THENC4=1ELSE
   C4=0
420 IFC2=3ANDC4=1THENX=25:GOTO510:ELSENEXTY:NEXTZ:GOTO480
430 IFPL()10THEN450ELSEC2=0:FORZ=1TO5:C1=R(Z):FORY=1TO5:IFR(Y)=
   C1THENC2=C2+1:IFC2)7THEN480
440 NEXTY:NEXTZ:X=30:GOTO510
450 IFPL()11THEN470ELSEC2=0:FORZ=1TO5:C1=R(Z):FORY=1TO5:IFR(Y)=
   C1THENC2=C2+1:IFC2)5THEN480
460 NEXTY:NEXTZ:C3=0:FORZ=1TO5:C3=C3+R(Z):NEXTZ:IFC3=15ORC3=20T
   HENX=40:GOTO510:ELSEGOTO480
470 IFPL=12ANDR(1)=R(2)ANDR(2)=R(3)ANDR(3)=R(4)ANDR(4)=R(5)THEN
   X=50:GOTO510
480 PRINT@L+320," SORRY !!  YOU";:PRINT@L+384," RECEIVE -1 !!";
   :FORZ=1TO1500:NEXT:IFPP=1THENA(PL)=-1:PP=2:ELSEB(PL)=-1:PP=1
490 K$=INKEY$:RETURN
500 REM                    (((( ASSIGN VALUES ))))
510 IFPP=1ANDA(PL)=0THENA(PL)=X:PP=2:ELSEIFB(PL)=0THENB(PL)=X:P
   P=1
520 T1(1)=0:T1(2)=0:FORZ=1TO6:T1(1)=T1(1)+A(Z):T1(2)=T1(2)+B(Z)
   :NEXT:IFT1(1))=63THENT1(1)=T1(1)+35
530 IFT1(2))=63THENT1(2)=T1(2)+35
540 T2(1)=0:T2(2)=0:FORZ=7TO13:T2(1)=T2(1)+A(Z):T2(2)=T2(2)+B(Z
   ):NEXT:T2(1)=T2(1)+T1(1):T2(2)=T2(2)+T1(2)
550 K$=INKEY$:RETURN
560 REM                    (((( DETERMINE ROLLS )))
570 FORZT=1TO26
580 RANDOM:R(1)=RND(6):R(2)=RND(6):R(3)=RND(6):R(4)=RND(6):R(5)
   =RND(6)
590 CLS:PRINT"           Y A H T Z E E":PRINT
600 PRINTN$(PP)"'S TURN.":PRINT"YOUR ROLL :"
610 PRINTSTRING$(26,176):FORZ=1TO5:PRINTCHR$(191)Z" ";:NEXT:PRI
   NTCHR$(191)"       DICE NUMBER":FORZ=1TO5:PRINTCHR$(191)STRI
   NG$(4,140);:NEXT:PRINTCHR$(191):PRINTCHR$(191)
620 REM 448
630 FORZ=1TO5:PRINT@D(Z),R(Z)" "CHR$(191);:NEXT:PRINT"       NUM
   BER ON DICE":PRINTSTRING$(26,131):IFXX=1THEN650
640 FORTR=1TO2
650 PRINT@576,STRING$(63,32);:PRINT@576,"HOW MANY DICE DO YOU W
   ISH TO RE-ROLL?";
660 K$=INKEY$:IFK$=""THEN660ELSED=VAL(K$):IFD)5THEN660
670 IFK$="L"THENXX=1:GOSUB140:GOTO590:ELSEXX=0
680 PRINT"  "D:IFD=0ANDK$="0"THENGOSUB140:NEXTZT:GOTO730:ELSEIF
   D=0THEN650
690 FORZ=1TOD:E(Z)=0:NEXT:FORZ=1TOD:PRINT@640+64*(Z-1),"WHICH D
   ICE DO YOU WISH TO RE-ROLL?";
700 K$=INKEY$:IFK$=""THEN700ELSEX=VAL(K$):IFX(1ORX)5THEN700ELSE
   FORY=1TOD:IFE(Y)=XTHEN700ELSENEXT:PRINT"   "X:E(Z)=X:PRINT@D
   (X),"X";
710 R(X)=RND(6):PRINT@D(X),R(X);
720 NEXT:PRINT@576,CHR$(31);:NEXTTR:FORZ=1TO500:NEXT:GOSUB140:N
   EXTZT
730 IFT2(1))T2(2)THENWW=1ELSEWW=2
740 ZZ=1:GOTO140
750 END:'OF PROGRAM
760 CLEAR100:DIMA(13),B(13),R(5):FORZ=1TO5:PRINT"DICE #"Z" :";:
   INPUTR(Z):NEXT:INPUT"POSITION :";PL:GOSUB360:PRINTX;C1;C2;C3
   :GOTO760
```

MASS/MAIL SYSTEM

This is the NAME and ADDRESS system for subscription control or large mailing lists. It will handle up to 10,500 records, with a worst access time of less than 15 seconds and usual access of less than one second. All adds, deletes and edits are instant for the operator and are then completed later in a "batch mode". Extensive documentation and ongoing support. Requires TRS-80 Model II and at least two disk drives. Contact 80-U.S. direct for detailed specifications and prices for your exact needs.

..... Model II Version only.....Contact us for price

MAIL/FILE SYSTEM

This is the name, address, phone number data base manager that has set the standard by which other systems are compared. This system contains advanced editing and output capabilities. The TRS-80 Model I sysstem will handle up to 600 records per file, while the Model III version will handle 2500 records per file. All versions are file compatible and maintain constant sort indexes on both NAME and ZIP code. International PHONE numbers and ZIP codes are supported. Thousands of code combinations are available. The Model III version also has a "word processor" type input editor and fast assembler sorting. Complete documentation is included with each version of MAIL/FILE.

Model I Version.....$ 99.00
Model II Version $199.00
Model III Version $149.00

EDAS 4.0 (Editor/Assembler)

This is the highly acclaimed "user oriented" assembler for the TRS-80 Model II by Galactic Software. Loaded with features such as assemble to memory, block move, link to debugger, default filenaming, reverse video editing, warm start entry, and much more. Now the programmer can write, assemble, test and debug his code without ever leaving EDAS.

EDAS 4.0 with complete Manual (120 pages) Model II only.....$ 179.00

STOCK MARKET MONITOR

This day to day market monitor is designed for the active trader. The system will track the performance of an issue against the market as well as against itself. The package comes with complete documentation and explainations of the formulas that are used by the program. The system is available for the Model I and the Model III TRS-80.

Model I and III Cassette Version.....$ 89.00
Model I and III Disk Version .....$ 99.00

80-U.S. SOFTWARE
PHONE (206) 475-2219

Circle 46

VISA

3838 South Warner Street
Tacoma, WA 98409

master charge

OMNI-KEY: The Utility for Mere Humans

Mere humans. Sounds insulting, doesn't it? But the fact is, our computers tower over us in one principal virtue. Patience. They can await input for days on end without becoming bored. They can digest DATA statement after endless DATA statement and not once complain of the tedium. They endure our most serious blunders with aplomb. And we humans? We curse the monotony of program entry, mutter at our clumsiness with EDIT, and rail at Tandy for their %#&!%! inadequate keyboard. *Aargh!* Computers are supposed to relieve this tiresome aggravation, not intensify it! Why doesn't somebody do something?

We have. We wrote OMNI-KEY. And if you had OMNI-KEY, your programming would not only be less tiresome, but more productive. How? Well, when was the last time these little annoyances got under *your* skin?

KKey BBounce. OMNI-KEY eliminates it.

Repetitive Keying of the Same Character. OMNI-KEY has autorepeat. Hold *any* key down, and it repeats about eight per second.

Typing Out Common Keywords. OMNI-KEY lets you assign BASIC keywords to the *SHIFT*-letter keys. Type *SHIFT*-P, for example, and you get PEEK(, or whatever *you've* made that key represent.

Repetitive Keying of Similar Phrases. Have you typed "DATA x, y, z *ENTER*" one too many times? OMNI-KEY's macro key types the repetitive stuff with a single keystroke. You just fill in the blanks. In fact, it's possible to enter hundreds of DATA statements in a row without typing line numbers, "DATA", or the commas! OMNI-KEY's unique macro pause and macro repeat make it possible. And *you* program the macro key any time and any way (up to 80 characters) you see fit.

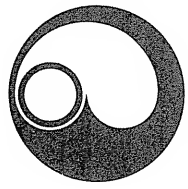
The "What's on the right of the cursor?" EDIT Mode. Come on. You don't *need* to put up with this half-blindness when editing a program. If the statement is listed on the screen (even a multi-liner), OMNI-KEY lets you edit it *in place and in full view* with its movable cursor. And you don't need any fancy commands to do it, either. To insert characters, just type them -- the lines will shift to accomodate them. Deletions are even easier -- just hit the *CLEAR* key. Need to move a statement? Just edit the line number! It's that easy.

Separate Drivers for Lower-case, Printers, Video Display, etc., etc. OMNI-KEY has its own lower-case driver and shifting built in. But the real beauty of the beast is what *you* can add to it. If you can use the Editor/Assembler, you can write your own OMNI-KEY modules. OMNI-KEY has a configuration mode which reads your specially-assembled SYSTEM tapes and merges them with the OMNI-KEY functions into *one single load module*. OMNI-KEY, in its standard and customized forms is equally at home with Level II or Disk BASIC, and you don't even have to set MEMORY SIZE to use it! Just enter BASIC, LOAD or CLOAD OMNI-KEY and RUN. It activates itself, reserves its own memory, and waits in the background until needed. Pretty simple.

Simplicity. That's the power of OMNI-KEY. It's simple, it's easy to live with, and it lets you, the programmer, do what you do best. Program. Without the tedium, without the aggravation, and, best of all, without spending a lot of bucks. At only $23, OMNI-KEY has got to be the best deal going! And if you're a mere human, that's something to think about.

OMNI-KEY *cassette for TRS-80 Model I, Level II and Disk BASIC, instructions, postpaid to any U. S., Canadian, or Mexican address. Others are F. O. B. Port Townsend. VISA and Mastercharge are welcome. Dealer inquiries are invited.*

$23

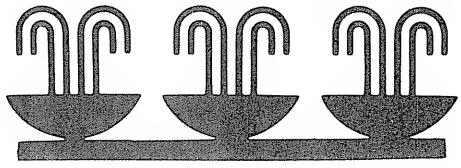


DISCOVERY BAY SOFTWARE CO.

Circle 47

P.O. Box 464 Port Townsend, WA 98368

Weir Flow



J. C. Dahlke
Adrian, Michigan

WEIR, a barrier of moderate height placed across a river or stream to control the flow of the water for navigation, irrigation, power, or other purposes, discharging the water over its crest or through wide openings having movable gates. River weirs are frequently termed dams, particularly in the United States where the term weir is more generally applied to small barriers erected to measure the flow or to trap fish. The French term *barrage* is often applied to weirs.

Encyclopaedia Britannica

For many engineers and technologists (geologists, water and waste plant operators, hydrologists, etc.) it is often necessary to estimate flows in flumes, channels, and from reservoirs. This Weir Flow Estimator was designed to do that.

This program is for water flows. Other liquids are metered or estimated by other means.

This type of calculation can be done on a hand-held calculator if evaluating one flow only, unless using a programmable calculator. This program will do single head evaluations. Further, it was designed to produce tabular results for a range of heads. The flows are presented by head, cubic feet per second, gallons per minute, and/or millions of gallons per day.

The display limitations are functions of weir length and head. It is possible to scramble the display by entering excessively large parameters. By the same token, it is readily possible to modify the program to handle larger parameters. The program was designed to cover about 80% of the weirs found in the United States.

Modifications should be done only if the user has some understanding of hydraulics. The program, as with all programs, should be parallel checked to determine suitability for application. This program does not allow for all variables that can be present in weir flow estimation. Therefore, it would be advisable to determine whether this approach would be appropriate for a given application. The program should not be used in applications involving property, health and life unless the user has determined the program to be suitable to the application.

Printer output routines have not been included with the program. Due to the variety of control techniques that are implemented with the TRS-80 systems and non-TRS-80 peripherals, it was felt that the user would be more familiar with the conventions of his own system. It has therefore been left to the user to provide his own printer routine, if desired.

This program is written in TRS-80 Level II Basic. It will run in 16K machines and up. It would probably run in 4K if the remarks were deleted and the code compressed.

Explanations for Weir Flow

Flow is an expression of quantity moving through a plane per unit of time. It is a rate.

A weir is a structure by which flow estimates can be made. The weir crest is level. There are four basic classes of weirs. 1) rectangular contracted, 2) rectangular suppressed, 3) Cipolletti, and 4) V-notch. The rectangular contracted weir does not extend fully across the channel being estimated, therefore the flow contracts at the ends of the weir. The rectangular suppressed weir extends fully across the channel being estimated, the flow does not contract at the ends of this weir. The Cipolletti or trapezoidal weir has definite geometric ratio of slope to the sides and compensates for the end contractions. The V-notch weir is conventionally used for low or metering flows. The V-notch normally uses the 90 degree angle. The rectangular contracted, Cipolletti, and V-notch weirs are the notch-type weirs - that is, they do not extend fully across the channel.

Length is used only on the first three classes of weirs. The V-notch weir does not consider length.

Head is an expression of pressure due to the weight of a height of water. Normally, head is expressed in feet. Head is measured from datum (point of reference - in this case from the crest of the weir). In considering a range of flows: a lower head is selected and proceeds incrementally to an upper head. Flow is exponential.

This type of evaluation expresses results in: cubic feet per second, gallons per minute, and million gallons per day.

There are many very good engineering handbooks which deal with weir flow in great detail.

```
10 '              IDENTIFICATION
20 '              WEIR FLOW ESTIMATOR
30 '     PROGRAM BY J.C. DAHLKE,   AUGUST 1980
40 '     2411 OCCIDENTAL HWY, ADRIAN, MI., 49221
50 '              VERSION 1.1
60 CLS: PRINTCHR$(23)
70 PRINT@340,"WEIR FLOW":PRINT@660,"ESTIMATOR": FOR TT%=0TO820:
    NEXT
80 CLS
90 '
100 '             INITIALIZATION
110 CLEAR 250
120 DEFINTF,N,X
130 C1=3.33: C2=2.5: K1=0.646317: K2=448.831
140 D0$="##.##": D1$="###,###.##": D2$="#####,###"
150 '
160 '             MENUS
580 CLS: PRINT@64,"HEAD","CFS","MGD","GPM": PRINT:  RETURN
590 CLS: PRINT@64,"HEAD","CFS","GPM": PRINT:RETURN
600 IF X=1 THEN PRINT"((ENTER)) INPUT FOR CONTRACTED RECTANGULA
    R WEIR": PRINT@960,"HEAD SHOULD NOT EXCEED HALF LENGTH";
610 IF X=2 THEN PRINT"((ENTER)) INPUT FOR SUPRESSED RECTANGULAR
     AND CIPOLLETTI WEIR": PRINT@960,"HEAD SHOULD NOT EXCEED HAL
    F LENGTH";
620 IF X=3 THEN PRINT"((ENTER)) INPUT FOR V-NOTCH WEIR": PRINT@
    960,"UPPER HEAD SHOULD NOT EXCEED 3 FEET";
630 IF X=3 GOTO 650
640 PRINT@192,"WHAT IS THE LENGTH OF THE WEIR IN FEET AND 1/10T
    HS";:INPUTL$: L=VAL(L$): IF L(=0 GOTO 640
650 PRINT@320,"EVALUATING (O)NE OR (R)ANGE OF HEADS";:INPUTR$
660 IF LEFT$(R$,1)="O" THEN GOTO 670  ELSE IF LEFT$(R$,1)="R" T
    HEN GOTO 680  ELSE GOTO 650
670  PRINT@320,"WHAT HEAD ARE YOU EVALUATING (IN FEET AND 1/10T
    HS)";: INPUT HA$: IF HA$="" GOTO 670  ELSE HA=VAL(HA$): IF H
    A(=0 GOTO 670  ELSE GOTO 750
680 PRINT@320,"LOWER HEAD (IN FEET AND 1/10THS)          ";: INP
    UT H2$: IF H2$="" GOTO 690  ELSE H2=VAL(H2$): IF H2(0 THEN 6
    80  ELSE IF H2=0 GOTO 690  ELSE GOTO 700
690 H2=0.1
700 PRINT@448,"TO UPPER HEAD (IN FEET AND 1/10THS) ";: INPUT H1
    $: H1=VAL(H1$): IF H2)H1 GOTO 680  ELSE IF H1(=0 GOTO 700
710 GOSUB 550
720 PRINT@576,"BY WHAT INCREMENT ";: INPUT S$: IF S$="" GOTO 74
    0  ELSE S=VAL(S$): IF S(0 GOTO 720  ELSE IF S=0 THEN S=0.1
730 GOTO 750
740 S=0.1
750 GOSUB 550  : RETURN
760 '
770 '          INSTRUCTIONS
780 CLS: PRINT:PRINT:PRINTTAB(5)"THIS PROGRAM ESTIMATES FLOWS F
    OR FOUR (4) TYPES OF"
790 PRINT"WEIRS.  THESE ESTIMATES ARE REASONABLY ACCURATE.  HOW
    EVER,"
800 PRINT"IF THE INPUT CONDITIONS WOULD RESULT IN EXTREME INACC
    URACY"
810 PRINT"OR DO NOT MEET STANDARD CONDITIONS, AN APPROPRIATE ER
    ROR"
820 PRINT"MESSAGE WILL BE DISPLAYED AND YOU WILL BE RETURNED TO
    "
830 PRINT"THE MENU."
840 PRINT:PRINT:PRINT:PRINTTAB(5)"THE PROGRAM WILL REQUEST YOUR
     SELECTION FROM A"
850 PRINT"MENU.  ONCE THE SELECTION IS MADE YOU WILL BE REQUEST
    ED"
860 PRINT"TO FURNISH CONDITIONS AND PARAMETERS."
870 GOSUB 540  : GOSUB 530  : CLS
880 PRINT: PRINT:PRINTTAB(5)"SIMPLE SELECTIONS REQUIRE A SINGLE
     KEYSTROKE ENTRY."
890 PRINT"PARAMETERS AND CONDITIONS WILL BE FOLLOWED BY PRESSIN
    G"
900 PRINT"(ENTER)."
```

```
170 CLS: PRINT@285,"MENU": PRINT@408,"1)  INSTRUCTIONS": PRINT@
    472,"2)  LIMITATIONS": PRINT@536,"3)  OPTIONS": PRINT@852,"S
    ELECT NUMBER OF CHOICE"
180 GOSUB 510
190  ON X GOTO 770  ,970  ,210
200 GOTO 160
210 CLS: PRINT@142,"WEIR SELECTION": PRINT@340,"1)  RECTANGULAR
    ": PRINT@404,"2)  CIPOLLETTI (TRAPEZOIDAL)": PRINT@468,"3)
    V-NOTCH (90 DEGREE)":PRINT@911,"SELECT WEIR TYPE"
220 N=0: S=0: HA=0: H1=0: H2=0: L=0: X=0: HA$="": H1$="": H2$="
    ": S$="": L$=""
230 GOSUB 510  : IF X=1 GOTO 240  ELSE GOTO 270
240 CLS: PRINT@215,"RECTANGULAR WEIRS": PRINT@348,"1)  CONTRACT
    ED (NOTCH)": PRINT@476,"2)  SUPRESSED (FULL WIDTH)": PRINT@5
    99,"SELECT TYPE"
250 GOSUB 520
260 '
270 '           CALCULATION & DISPLAY RESULTS
280 CLS: GOSUB 600
290  IF LEFT$(R$,1)="0" ON X GOTO 1220 ,1230 ,1240
300 M=H2
310 FOR HA=H2 TO H1+S/2 STEP S: IF N=0 ON X GOSUB  580  ,580  ,
    590
320 HH=HA
330 N=N+1: M=M+S
340 ON X GOSUB 420  ,450  ,480
350 IF N=10 AND M()H1+S AND HH(H1 GOSUB 530
360 IF N=10 THEN N=0
370 NEXT
380 GOTO 1250
390 '
400 '             SUBROUTINES
410 '             RECTANGULAR CONTRACTED WEIR
420 Q=C1*(L-.2*HH)*(HH^1.5): Q1=Q*K1: Q2=Q*K2
430 PRINT USING D0$;HH,: PRINT, USING D1$;Q,: PRINT, USING D1$;
    Q1,: PRINT, USING D2$;Q2: RETURN
440 '             RECTANGULAR SUPRESSED & CIPOLLETTI WEIRS
450 Q=C1*L*(HH^1.5): Q1=Q*K1: Q2=Q*K2
460 PRINT USING D0$;HH,: PRINT, USING D1$;Q,: PRINT, USING D1$;
    Q1,: PRINT, USING D2$;Q2: RETURN
470 '             V-NOTCH WEIR
480 Q=C2*(HH^C2): Q2=Q*K2
490 PRINT USING D0$;HH,: PRINT, USING D1$;Q,: PRINT, USING D1$;
    Q2: RETURN
500 RETURN
510 IK$=INKEY$: X=VAL(IK$): IF X(1 OR X)3 THEN 510  ELSE RETURN
520 IK$=INKEY$: X=VAL(IK$): IF X(1 OR X)2 THEN 520  ELSE RETURN
530 IK$=INKEY$: IF IK$="" THEN 530  ELSE RETURN
540 PRINT@979,"PRESS ANY KEY TO CONTINUE";: RETURN
550 IF (X=3 AND  HA)3) GOTO 1180 ELSE IF (X=3 AND H1)3) GOTO 11
    80 ELSE IF X()3 AND (H1-H2)L/2 OR HA)L/2) GOTO 1190 ELSE RET
    URN
560 CLS: PRINT@512,"HEAD","CFS","MGD","GPM": PRINT: RETURN
570 CLS: PRINT@512,"HEAD","CFS","GPM": PRINT: RETURN
910 PRINT: PRINT:PRINTTAB(5)"THIS PROGRAM DOES NOT MAKE ESTIMAT
    ES FOR ALL"
920 PRINT"CONDITIONS AND PARAMETERS.  ONLY STANDARD CONDITIONS"
930 PRINT"ARE CONSIDERED.  IF INPUT CONDITIONS INDICATE A NON-"
940 PRINT"STANDARD WEIR EVALUATION, YOU SHOULD CONSULT YOUR"
950 PRINT"ENGINEERING HANDBOOK.":GOSUB 540  :  GOSUB 530  : CLS
    : GOTO 160
960 '
970 '           LIMITATIONS
980 CLS: PRINTTAB(20)"LIMITATIONS"
990 PRINT:PRINTTAB(5)"THE FRANCIS FORMULA SEEMS TO AGREE WITH E
    XPERIMENT"
1000 PRINT"WITHIN 3% (RECTANGULAR AND CIPOLLETTI WEIRS) WHEN:"
1010 PRINT:PRINTTAB(7)"1)  LENGTH (L) OF WEIR IS GREATER THAN 2
     TIMES"
1020 PRINTTAB(12)"HEAD (H)"
1030 PRINTTAB(7)"2)  HEIGHT OF WEIR CREST IS ABOVE THE BOTTOM O
     F THE"
1040 PRINTTAB(12)"CHANNEL AT LEAST 3 TIMES HEAD (H)"
1050 PRINTTAB(7)"3)  HEAD (H) IS NOT LESS THAN 0.3 FEET"
1060 PRINTTAB(7)"4)  HEAD (H) SHOULD BE MEASURED AT LEAST 4 TIM
     ES"
1070 PRINTTAB(12)"HEAD (H) FROM THE WEIR"
1080 PRINT:PRINTTAB(5)"IN A CASE WHERE HEAD (H) IS GREATER THEN
     LENGTH"
1090 PRINT"(L) RELIABILITY STRAYS TOWARD 7%.": GOSUB 540  : GOS
    UB 530  : CLS
1100 PRINT:PRINT:PRINTTAB(5)"V-NOTCH WEIRS ARE GENERALLY USED F
    OR ESTIMATING"
1110 PRINT"FLOWS WHERE HEAD (H) IS 3 FEET OR LESS."
1120 PRINT:PRINT:PRINT:PRINTTAB(5)"IF IN DOUBT, CONSULT AN ENGI
    NEERING  HANDBOOK,"
1130 PRINT"STEVENS WATER RESOURCE BOOK,  OR OTHER TEXT ON APPLI
    ED"
1140 PRINT"HYDRAULICS."
1150 PRINT: PRINT: PRINT:PRINT"THIS PROGRAM IS INTENDED FOR EST
    IMATING ONLY.": GOSUB 540  : GOSUB 530  : CLS: GOTO 160
1160 '
1170 '                  ERROR MESSAGES
1180 CLS: PRINT@521,"EXCEEDING LIMIT OF 3 FOOT HEAD": GOSUB 540
      : GOSUB 530  : GOTO 160
1190 CLS: PRINT@512,"NON-STANDARD EVALUATION": GOSUB 540  : GOS
    UB 530  : GOTO 160
1200 '
1210 '           SINGLE EVALUATION AND DISPLAY
1220 GOSUB 560  : Q=C1*(L-.2*L)*(HA^1.5): Q1=Q*K1: Q2=Q*K2:PRIN
    T USING D0$;HA,: PRINT, USING D1$;Q,: PRINT, USING D1$;Q1,:
    PRINT, USING D2$;Q2: GOTO 1250
1230 GOSUB 560  : Q=C1*L*(HA^1.5): Q1=Q*K1: Q2=Q*K2:  PRINT USI
    NG D0$;HA,: PRINT,  USING D1$;Q,: PRINT,  USING D1$;Q1,: PRI
    NT, USING D2$;Q2: GOTO 1250
1240 GOSUB 570  : Q=C2*(HA^C2): Q2=Q*K2: PRINT USING D0$;HA,: P
    RINT, USING D1$;Q,: PRINT, USING D2$;Q2: GOTO 1250
1250  PRINT@960,"ANY FURTHER EVALUATIONS?";: GOSUB 530  : IF LE
    FT$(IK$,1)="Y" GOTO 210  ELSE CLS:END
```

Apparat, Inc.

4401 South Tamarac Parkway
Denver, Colorado 80237

Announces New Products for: TRS-80® Apple® Atari®

HARDWARE

Disk Drives
with power supply and chassis

40 Track

Tandon or Shugart single sided	**$339.00**
Tandon TM100-2A dual sided	**$499.00**

80 Track

Tandon TM100-3 single sided	**$679.00**
Tandon TM100-4 dual sided	**$839.00**

Apple Users
Apparat introduces an add on dual sided 40 track drive that more than doubles the storage capacity of your disks.
With necessary software **$675.00**

Coming Soon
5 1/4" Winchester hard disk drives

THE APPARAT CONNECTION
Provides a direct connect modem for interfacing a computer through a telephone line to another device. The CONNECTION plugs into a standard telephone extension jack. An additional RS-232 plug is available to connect to a printer.
Connects through TRS-80® Bus or RS-232 **$249.00**
RS-232 Only **$199.00**

Connects through Atari® Bus **$249.00**

Auto dial option for either TRS-80 or Atari **$79.00**

Smart Terminal 80 (Disk), one of the most powerful Smart Terminal programs ever written for the TRS-80®, is designed to work especially well with the CONNECTION. **$79.00**

Clock/Calendar/Appointment Scheduler
Provides the usual clock/calendar functions including hour/minute/second and month/day/year/ day of week; however, here the similarity ends. The user is given a choice of two output modes, either the user's program can request the time and date or these values can be DMAed to the computer where they are automatically displayed on the video screen. Also included is the capability of setting nearly a hundred time/dates which, when matching the time/ date of the clock/calendar, will cause an interrupt. A Basic program included can be used to maintain an appointment schedule for one or more people or alert persons to perform certain tasks.
Available in March for TRS-80® and Apple® **$169.00**

PROM Programmer
Apparat persents a PROM programmer that will handle all 25XX and 27XX parts in the 1k to 4k range and both 5v only and multivoltage types. This is done by use of a 16 pin personality jumper plug which configures the programmer to the appropriate PROM type. Zero insertion force sockets are provided for each PROM type to read, write, edit and verify from/to any range of memory. Only a portion of a PROM need be written so that it may be programmed even if it is not a blank.
Available in February for TRS-80® and Apple® **$149.00**

SOFTWARE

"A complete Small Business Accounting System"
The **M**icro **A**ccounting **S**ystem General Ledger, Accounts Payable, Accounts Receivable are fully integrated with all entries automatically posted to all modules. The system will handle a General Ledger chart of 200 accounts, 300 vendors for Accounts Payable and 300 to 1600 customers under Accounts Receivable.

$159.00 Each for General Ledger, Accounts Receivable, or Accounts Payable
$ 89.00 Check Register
$489.00 Special for the entire system

"AT LAST a manufacturing control system"
Easy to use modular construction allows interfacing with Invoicing and Accounts Receivable. Manages inventory, sales and receivables with ease. Comfortably handles inventory as products comprised of components contained in the inventory. The system consists of the following modules which may be purchased separately:

	Model I	Model II	Apple
Invoicing and Sales Subsystem	**$175.00**	**$250.00**	**$175.00**
Manufacturing Inventory Control Subsystem	**$175.00**	**$250.00**	**$175.00**
Accounts Receivable with Labeling Subsystem	**$200.00**	**$300.00**	**$200.00**
Integrated Sales and Inventory System	**$500.00**	**$700.00**	**$500.00**

FLEXTEXT
An appendage to SCRIPSIT for the Model I, that allows all of SCRIPSIT formats while exercising many of the features of the **Centronics 737** printer such as:

- Proportional character set in normal and expanded mode.
- Right Justification using proportional or compressed character sets.
- Underlining
- Superscripts and subscripts **$29.95**

Diskettes Special - Quantities of 100 for **$229.00**
(Contact for Dealer pricing)

ASAP - (Apparat's Scientific Arithmetic Processor)
TRS-80 $399.00 **Apple $349.00**

Epson MX-80 with cable	**$645.00**
Centronics 737	**$839.00**
Anadex DP-9500 or DP-9501	**$1499.00**
NEC Spinwriter	**$2449.00**
NEW DOS/80	**$149.00**

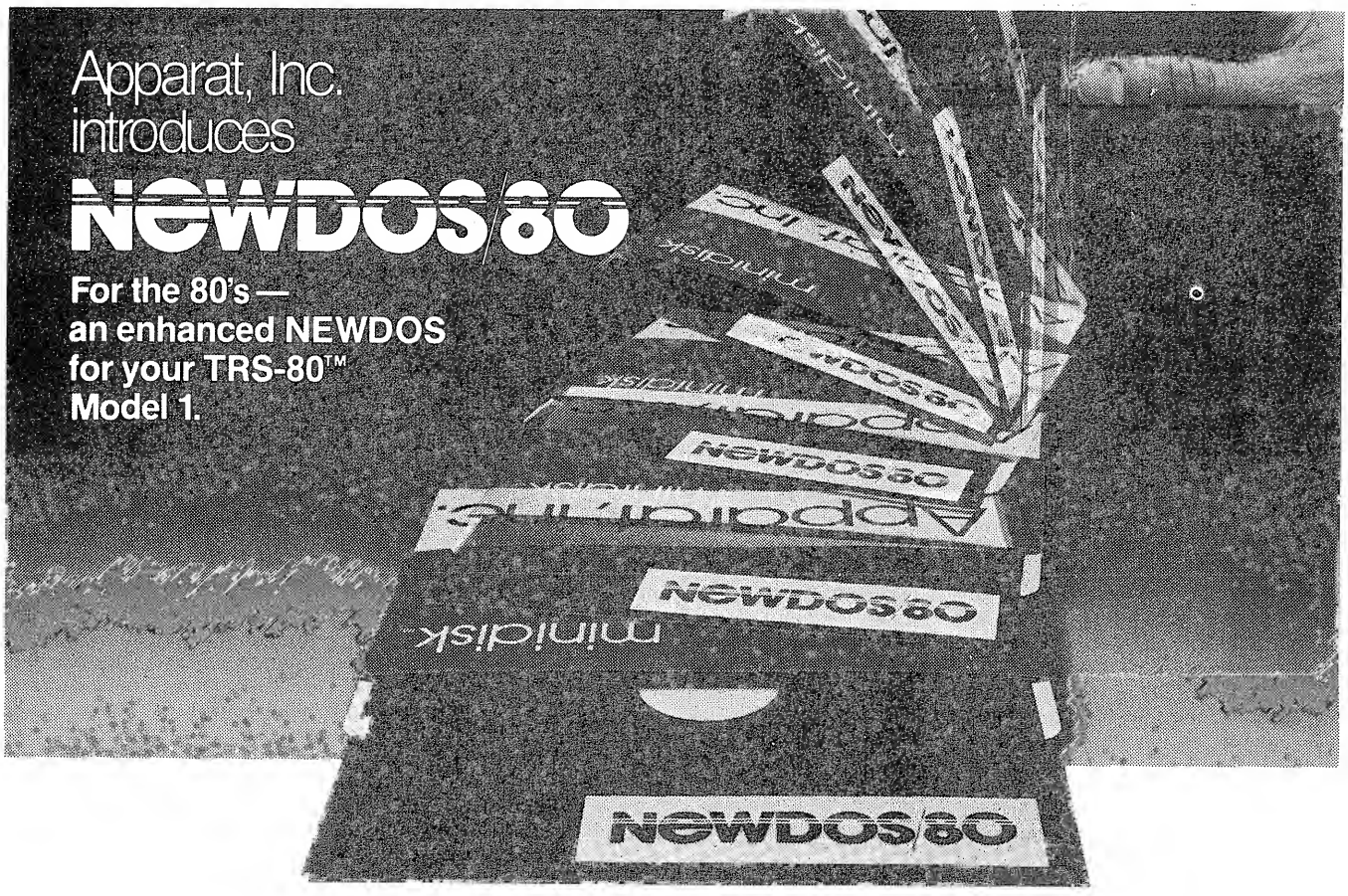
Call us for your software needs for TRS-80, Apple or CP/M. Complete selection of North American Software, the most sophisticated business package for the TRS-80 Model II.

ASK FOR OUR
NEW FREE CATALOGUE

Circle 49

Order Desk 1-(800)-525-7674 (except Colo.)
Technical/Service 1-(303)-741-1778

Minimum Visa or Master Charge Order $10.00
All Cash Purchases Discounted Freight FOB Denver



Apparat, Inc., announces the most powerful Disk Operating System for the TRS-80®. It has been designed for the sophisticated user and professional programmer who demands the ultimate in disk operating systems.

NEWDOS/80 is not meant to replace the present version of NEWDOS 2.1 which satisfies most users, but is a carefully planned upward enhancement, which significantly extends NEWDOS 2.1's capabilities. This new member to the Apparat NEWDOS' family is upward compatible with present NEWDOS 2.1 and is supplied on Diskette, complete with enhanced NEWDOS + utility programs and documentation. Some of the NEWDOS/80 features are:

- New BASIC commands that supports files with variable record lengths up to 4095 Bytes long.
- Mix or match disk drives. Supports any track count from 18 to 80. Use 35, 40 or 77 track 5" mini disks drives or 8" disk drives, or any combination.
- A security boot-up for BASIC or machine code application programs. User never sees "DOS READY" or ">READY" and is unable to "BREAK", clear screen, or issue any direct BASIC statement including "LIST".
- New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
- Enhanced and improved RENUMBER that allows relocation of subroutines.
- Powerful chaining commands.
- Print Spooler.
- DFG function; simultaneous striking of the D, F and G keys will allow the user to enter a mini-DOS to perform some DOS commands without disturbing the resident program. (e.g. dir while in scripsit.)
- Upward compatible with NEWDOS 2.1 and TRSDOS 2.3.
- Includes machine language Superzap/80 and all Apparat 2.1 utilities.
- Enter debug any time by pressing 123 keys. Also allows disk I/O.
- Diskette "Purge" command.
- Specifiable system options (limited sysgen type commands).
- Increased directory capacity.
- Copy by file commands.

NEWDOS/80 with all of the NEWDOS + utility programs, many of which have been enhanced, is priced at just $149.00 and is available at most TRS-80 dealers.

As with 2.1, NEWDOS/80 relies on the TRSDOS and Disk Basic Reference Manual published by Radio Shack. NEWDOS/80 documentation supports its enhancements and upgrades only.

NEWDOS/80
FOR THE TRS-80
MODEL III
AVAILABLE IN THE
SPRING 1981

TO PURCHASE NEWDOS/80, COMPLETE AND MAIL TO:

Apparat, Inc.
4401 S. Tamarac Parkway
Denver, CO 80237

CALL TODAY
Order Desk 1-800-525-7674 (Except CO, AK & HI)
Customer Service 1-303-741-1778

☐ Check ☐ Money Order ☐ Master Charge ☐ Visa

Card No _____ Expiration Date _____

Colo residents add 6.5% sales tax.
Outside Continental U.S. residents add $10 for Postage & Handling

Please rush _____ NEWDOS/80 @ $149 EACH TO:

Name _____

Address _____

City _____ State _____ Zip _____

Phone _____

Circle 50

ESFOA

EXATRON STRINGY FLOPPY OWNERS ASSOCIATION

Secretary, Fred Waters

WHAT IS IT?

ESFOA is a voluntary association of all individuals who own an Exatron Stringy/Floppy mass storage subsystem, or who are sufficiently interested in the ES/F to participate in ESFOA activities. There is no formal organization, nor are there any requirements or dues. The degree of participation is entirely up to each individual.

PURPOSES

The purposes of ESFOA are: to facilitate the interchange of ideas, information, software, and hardware design; to encourage mutual assistance and cooperation among members with specific common interests; and to furnish the administrative backup to support these purposes.

HOW DOES IT WORK?

1. There is a permanent secretary for administrative backup.
2. Local workshops are the primary means of carrying out the ESFOA purposes.
3. For those living near Exatron in Sunnyvale, there is a workshop meeting at the plant every Saturday morning from 9 to 11 AM. These have been continuous since January 1978. There is always coffee on hand, a round table discussion of whatever comes up, and friendly help from other members when needed.
4. There are hundreds of Workshop Representatives located throughout the country who can organize local ESFOA chapter workshops for those who are interested.
5. A newsletter is published periodically for distribution to all members.
6. Each interested member fills out an "Activity/Interest Record" which is used, as the occasion rises, to put members with common interests in touch with each other.
7. For those members interested in offering professional caliber programs for commercial distribution, ESFOA musters the assistance and support needed to sell such programs to other ES/F owners. Authors receive generous royalties from programs sold.

WHY AN OWNERS ASSOCIATION?

Computer owners are a tremendously diversified group of people. Their needs and specific interests vary widely. The largest element they have in common is that most of them are new to the field and have a large amount of learning to do. The second element in common is that most of them desire better software than is yet available, especially for applications purposes. Almost all people can help meet their own needs best by cooperating with other owners who have like interests.

To some degree all computer hobbyists are becoming programmers. By working together, they can write, revise, field-test, use, and sell many types of software. Among these types are: better software to use the ES/F hardware, utility routines, data storage techniques, monitors, and others. Business applications programs are also greatly needed.

ESFOA operates on a policy of giving software credit where it is due. All members are encouraged to write software. Where the author wants to market his work, ESFOA affords him as much protection as possible. Other authors believe in donating smaller works to the public domain. In either case, the author gets the credit and whatever else he is entitled to. ESFOA follows the same policy on software from any source and will not sanction unauthorized copying of proprietary software.

HOW HAS IT WORKED SO FAR?

Past events show why an owners association can and will be successful. When the S-100 version of the Exatron Stringy/Floppy was first offered, the only software available was a rudimentary utility routine, just enough to make the hardware operate. That was a long way from the highly useful storage system that now exists with its accompanying operating software and utility routines.

Similarly with the S-80 version, a number of individuals brought together through the ESFOA workshop meetings, have developed much of the hardware and software now available. And the task is by no means finished. New and improved software of many types is still needed. In the task of creating better software, the more experienced programmers have, on many occasions, helped solve software problems for those with less experience.

Every workshop meeting has provided answers to questions, particularly with newcomers. Tutorials have been of great interest to virtually all persons attending. And the wide variety of problems solved has included many which are not related to the ES/F itself but rather to printer interfacing, driver routines, worthwhile software, and hundreds of other learning problems that each person faces at one time or another.

EXPERIENCE DEMONSTRATES THAT THIS IDEA WORKS!

In accordance with that experience, the Exatron Stringy/Floppy Owners Association will continue to encourage all kinds of cooperation among its members and will provide services that the ever-expanding body of members will find useful.

THE EXATRON STRINGY FLOPPY OWNERS ASSOCIATION NEWSLETTER

What is an "Exatron Stringy Floppy?"

For those of you who are not yet familiar with the Exatron Stringy Floppy please read our advertisement on the back cover of this magazine. A complete information package will be mailed to you on request.

What is the "Exatron Stringy Floppy Owners Association (ESFOA)"?

A copy of the ESFOA charter is reproduced on the opposite page for the benefit of prospective new members.

What is an ESFOA "Workshop"?

An ESFOA workshop is similar to a computer club. The main difference is that computer clubs are strictly local groups of computer owners whereas Exatron Stringy Floppy (ESF) owners are located all around the globe and workshops are being organized everywhere when enough interested ESFOA members can be found.

Workshops provide a place where owners can interact with each other to accomplish useful "work". They also provide a place where prospective new members can go to see the Exatron Stringy Floppy demonstrated and learn about the benefits of membership in the ESFOA.

What is an ESFOA "Workshop Representative"?

An ESFOA Workshop Representative is an ESF owner who has volunteered to answer questions, help new owners, and give demonstrations to prospective owners. He is also a person who is interested in helping to organize an ESFOA "Workshop" in the local area when there are enough members to make regular meetings worthwhile.

The ESFOA supplies the name and address of new owners and prospective new owners to the Workshop Representative so that he can contact them to organize a workshop. There are presently over 400 Workshop Representatives and the number is growing daily.

What is the purpose of the ESFOA Newsletter "@NEWS"?

The purpose of the ESFOA Newsletter @NEWS is to provide a method of communication among the ESFOA, its members, Workshop Representatives, and prospective new members.

The ESFOA has been in existence for three years and regular workshops have been held at the Exatron plant from 9 to 11 AM every Saturday morning. During this period of time Exatron has moved its location three times. Each move has been to enlarge the facilities by a factor of three. In the meantime the number of ESFOA members has grown from the very first one to nearly 5000. This number will soon exceed 10,000 and the problem of keeping every member informed about activities and other news has grown accordingly.

The purpose of this newsletter, therefore, is to allow members to share solutions to problems they have found and, in general, to allow owners to obtain information which will increase the usefulness of the Exatron Stringy Floppy as a mass storage subsystem for their computer system.

Why is the ESFOA Newsletter printed in the 80 US Journal

First of all the writing, editing, printing, and mailing of a newsletter is a big job which requires professional expertise. As you might guess, undertaking such an operation can be very expensive and it would be impossible to mail a "free" copy to all members.

After a careful evaluation of all the other computer magazines, the 80 US Journal was selected as the most suitable media because of the high professional caliber of the articles published. Since the subscription rate for 80 US Journal is justified by other material published, the @NEWS newsletter will be a "free bonus" to ESFOA members.

How does the ESFOA newsletter @NEWS "work"?

The only way this newsletter can "work" is for members to submit ideas and suggestions which they would like to share with other members. Send your ideas and suggestions to @NEWS in care of ESFOA, 181 Commercial, Sunnyvale, CA 94086.

What else does the ESFOA do?

The ESFOA distributes computer programs on wafer for ESFOA members. Many excellent programs are currently available and a listing of these programs are contained in the ESFOA Software Catalog. As new ESF software becomes available, we plan to keep members informed by timely announcements in @NEWS.

Where does the ESFOA obtain computer programs?

Arrangements have been made with some prominant software companies to distribute several of their better programs in the ESF wafer format. These quality programs are also available in their original cassette or disc versions through other distributors. The ESFOA also acts as a publisher for software authored by ESFOA members who have the desire and talent necessary to produce software of professional quality. This is the source of many excellent programs and the list of successful ESFOA software authors is growing constantly.

What about "Public Domain Software"?

There is a large pool of public domain software which is available free. One of the challenging problems which the ESFOA is attempting to solve is how to collect, copy, and distribute this software to members who would like to obtain copies of these programs. The first step is to obtain copies of these "public domain" programs. To this end all members are encouraged to send in copies of worthwhile public domain software on ESF wafers.

If you have such a program, please give us a call here at Exatron and ask for the ESF Software Division. Tell them you want information on how to donate programs to the ESFOA library. The ESFOA would appreciate very, very much any help you can give in this regard. These programs will be cataloged and made available to all members for a very nominal handling charge. We feel this service will be an enormous benefit to many members.

What Business Software is available?

The most universal type of software which can benefit almost every business is "Word Processing". The Exatron Stringy Floppy is ideal for this purpose. The cost of a TRS-80 with an Exatron

Stringy Floppy and a suitable printer with Electric Pencil or Scripsit could be justified by many businesses for word processing use only.

What about other business applications?

There are many potential applications for computers in small businesses. The classic ones are general ledger, accounts receivable, accounts payable, inventory, and payroll. The dominant characteristic of all these applications is the need to handle "data files". A large number of TRS-80 owners have found that the audio cassette is not practical for such applications. The Exatron Stringy Floppy on the other hand has excellent "data file" handling capabilities and can be used successfully by many small businesses for one or more of the "classic" business applications. Future issues of @NEWS will devote much space to "data file handling" and practical examples of real applications will be used whenever possible.

If you have written a small business program that would handle one or more of these "classic" applications and would like to donate it to the ESFOA library, then again we strongly urge you to get in touch with the Software Division here at Exatron for information on procedures for donating programs.

It may be that you have written some business software that is of such a scope and professional caliber that it would merit consideration to be published through Exatron's software royalty system. If that be the case, then by all means get in touch with our Software Division for particulars on submitting software of commercial value for review.

What about the use of the Exatron Stringy Floppy in business programming?

Clearly an outstanding feature to have available in any computer system is the ability to link two or more programs together so that the variables and array data from one program may be carried through and utilized by a succeeding program. This technique of preserving and using variables data from one program to another is called "program chaining". The advantages of having this feature available in a business programming environment should be readily apparent. Unfortunately this chaining ability is not part of the TRS-80 Level II ROM and it has remained mostly just a wishful thought in the minds of most Level II Basic programmers.

The Exatron Stringy Floppy has now changed all this. The ability to "chain" programs is one of the many new programming advantages ESFOA members realize they now have.

To utilize this powerful programming tool is unbelievably simple and is thoroughly covered in the ESF Owners Manual on page 3.4. Briefly, the programming sequence goes like this:

Make sure that the first program file that is run is the longest physical file that will be run during your program file calling sequence. If it is not possible to have your first program be the longest, there is a way around this dilemma and I will discuss what to do about it in a moment.

The key to the whole operation is what is located in RAM memory addresses 16633 and 16634. These addresses contain the pointer to the beginning of the stored variables list, or, as some prefer to identify the contents, a pointer to the end of the resident Basic program. This pointer gets set when a program is loaded and remains intact until a new program is called from a typed in keyboard command such as @LOAD 2. (All Stringy Floppy commands are preceded by the "@" symbol.) When a program is called up in this manner, not only does that pointer get reset but all stored variables are lost also.

Now here is where Stringy shows his stuff. If a program is called up via a program line statement, e.g., 100 @LOAD 2; the address pointer in 16633 and 16634 does not get reset and the existing stored variables will not be clobbered by any variables set by the new program. Of course it becomes obvious that the called in program must not be longer than the original or you will get an OM error because of the preset pointer contained in 16633 and 16634. With the new program called up in this manner; it overlays the previous Basic program in memory but the stored variables remain undisturbed.

Now what do we do if the first program we must run in our sequence cannot be the longest program and thereby properly set that "key" pointer? Simple! Fool it. Make the computer think its loaded in a longer program than it really has. This is how it's done.

First load in the longest program of the sequence of programs you must run. After is has loaded, take a peek at those two addresses 16633 and 16634 and make a note of what their contents are. For example, let's say addresses 16633 and 16634 contained 240 and 82 respectively. Now insert the following statement as the first line of the first program that must be run:

```
10 POKE 16633,240: POKE 16634,82
```

Make this statement a permanent part of that first program and now every time it executes, it will automatically reset those pointers to a value where the computer thinks it has a program in memory that is as long as the actual longest one he will call up a little later on in the sequence.

There, that's all there is to it. Exatron Stringy Floppy has made it that simple. If any of you are into using Level III by Microsoft, or are planning to order it, you might want to order the Level III demonstration program that is available through the Exatron Commercial Software Division. This program uses "chaining" throughout six program files on a 50-foot wafer. This set of program files is an excellent example of using the "chaining" feature. Also, this same program (Level III DEMO) gives an example of another powerful feature that the Exatron Stringy Floppy has, and that is the ability to load program data directly into screen memory. This is really a neat feature and we'll be discussing its use in a later @NEWS.

What about the use of the ESF in schools?

Many schools are starting "Computer Literacy" classes. Most of these schools are using personal computers such as the TRS-80, Apple, or Pet with cassettes for mass storage. As most ESFOA members know, audio cassettes are really not suitable for this purpose. On the other hand, the speed, convenience, reliability, and low cost of the Exatron Stringy Floppy make it the ideal alternative for this application.

Since very few teachers read computer magazines, most of them are not aware of the Exatron Stringy Floppy and how it can help them achieve their "computer literacy" goals in the classroom. Because of this situation, Exatron will be making a major effort to bring the ESF to the attention of school teachers throughout the coming year. ESFOA members, and particularly Workshop Representatives, will be called upon to give demonstrations and help teachers who are starting computer classes in schools in their area. If you are interested in demonstrating the unique capabilities of the ESF to people in the educational field, please get in touch with us here at Exatron.

What about ESFs for "other computers"?

Yes, Exatron makes Stringy Floppies for many types of computers, and yes they are working on models for the Tandy Color Computer and the Model

III. There are ESFs for the Pet and Apple as well as an RS-232 model for just about every other kind of small or large computer.

Exatron is also working on a "hardware upgrade package" to greatly expand the performance and usefulness of your Model I TRS-80. There will be more on this in later newsletters.

How can I start my own computer business?

Exatron does not sell through retail dealers in the US although we do handle international sales through distributors located in most foreign countries. Nevertheless, it is Exatron's long-term goal to become a supplier of "packaged small business computers". The plan is to sell these products through "Direct Sales Representatives" who are, in effect, computer consultants serving their local area.

The first step towards becoming a Direct Sales Representative is to become an ESF owner with the second step following close behind, being a Workshop Representative. The third step is to organize a workshop and hold regular meetings. This sequence of steps will prepare you for the task of presenting your ideas and proposals before individuals and groups of business and educational-minded people.

To assist you in the task of running your own small business as a "Direct Sales Representatives", Exatron is in the process of preparing a series of correspondence courses. The prospective "Sales Representative" would be required to complete these courses before being authorized to act as a Direct Sales Representative for Exatron. This is an exciting concept that we feel will open up many career possibilities in all the computer-related fields.

What will future newsletters be like?

We felt this first issue of @NEWS should be rather general in nature. We wanted it to be sort of an introduction of the ESFOA to the new members as well as a general interest article to those old timers who have been reading 80 US Journal for a long time and now see a new kid on the block trying to muzzle in on some of the space of their favorite computer magazine. Who knows, when they find out what great guys and gals we ESFOA types are, we might make a few converts along the way.

Future issues of @NEWS will address more specific issues and questions regarding the operation and programming capabilities of the ESF. Some of the continuing articles we intend to feature will be such headings as:

1. ESFOA library program updates
2. Commercial Software Catalog updates
3. New Exatron products
4. Program listings and information on an idea called "Wafer of the Month" (WOM).
5. A feedback column for our members.
6. Listings of charter members of new Workshops as they come into being.
7. A "Share the Wealth" column in which we will disseminate useful and interesting facts that are passed along to ESFOA from Workshop meetings around the globe.

If you have any questions about Exatron, the Exatron Stringy Floppy or the Exatron Stringy Floppy Owners Association, please call the Exatron Hotline 800-538-8559 or within California 408-737-7111.

REVIEW XEDIT

Mi-Prog
P.O. Box 27014
Minneapolis, Minnesota
$44.95

Sophisticated editors are common these days, but usually you pay a price in order to have one. The price is a large system. Most editors are also designed for word processing. For many this is an advantage, but for some it is an unnecessary bother. Why have all that justification, paragraphing, and other software when all you need is just a good flexible editor?

MI-Prog of Minneapolis, has come out with the XEDIT editor for programmers. This is not a word processing editor. You could use it for such at about the same level as you use the Basic editor from Level II. You can even strip off the line numbers from a file if you want.

However, let's get back to the point. This is a programmer's tool primarily. Much like Microsoft's EDIT80, but at a much lower price for comparable features. More than that, XEDIT is part of a set of programs for the programmer that are compatible within the set as well as being compatible with such programs as the EDTASM program with NEWDOS and BASCOM, F80, M80, and other programs from Microsoft.

Versatility makes this an ideal editor for many of a programmer's needs. I've tried the editor on the Model I (Model II and III versions are also available). I found that the editor has a command syntax close enough to the Level II Basic to make learning it rather easy. Yet, it has some powerful enhancements that make editing much easier.

The editor is a line oriented, disk based editor. The current line for editing is always indicated by an internal pointer. Using some of the editor's commands, the pointer can be moved up or down through the file to whatever place desired. Most commands also move the pointer as a normal part of their operation.

XEDIT makes use of a sophisticated, yet simple "entry point table" technique to keep track of which lines belong where in the file. This powerful technique makes it a fast system since there is no need to make space in a file or otherwise get rid of space in a file when adding or deleting lines.

For the programmer, this translates into greater speed of operation. Further, the programmer need never worry about what line number the pointer is on, since most of the editing is handled very efficiently with search and replace commands.

The editor is capable of handling files up to 2.7 million bytes in size or up to 10K lines in length. This is an immense file size, larger than most programmers will need in ten lifetimes.

The commands in the system include:

Auto line numbering
Text insertion
Line replace
Delete lines
Edit lines
Locate string anywhere in the file
Change a string anywhere in the file
Move the pointer
Print the text
Print hard copy
Block move or copy text
Display pointer status
Write file
Disk directory
Quit or abort editor
Kill disk file
Set pointer forms parameters
Set text window

Just listing the various things the editor can do doesn't completely describe the capabilities though. For example, the insert lines mode is not just for inserting lines from the keyboard. It can also be used to insert text from a disk file.

I haven't reached the end of what can be done with the system by any means. It is such a versatile programmer's tool that it could easily replace other programs such as the editor portion of the EDTASM.

The greatest advantage to the system is that it was designed to provide compatibility with EDTASM and EDIT80 file structures. You can select what format the file will write to disk so that you can create a file with XEDIT, assemble it with EDTASM or M80 if you want, or just put the file into ASCII form on disk.

For testing with the editor, I created some short machine language program files and assembled them with Mi-Prog's ASM assembler. The whole system worked quickly and efficiently and I was rather impressed.

I guess my only complaint in testing the system is that I have become very "screen oriented" when it comes to editors. Much of my programming is done on screen editors of various sorts. Even when I'm working on a program for a large CDC Cyber machine, I often down load the program to my TRS-80 to make changes locally on my screen editor.

MI-Prog's editor can handle the downloaded programs without trouble since it will handle straight ASCII files, but is *not* a screen oriented editor Surprisingly, I found that the ease with which I was able to edit long program files made me almost forget my dependence on screen editing (remember, I said almostl).

In summary, I can safely say that during the period I used the XEDIT editor regularly for testing, I found it both easy to use and reliable. At no time during the testing period was I inconvenienced except through my own failure to follow the clearly written instructions.

I often attempted things that shouldn't really have been attempted, but none of them got me into serious trouble. Since the syntax is close to the Basic editor provided in the Microsoft interpreter, I often tried to use commands that I was familiar with. In some cases they didn't work the way I expected. For example, printing lines to the screen doesn't work the same as it does in EDTASM or the Basic editor. It's clear, just not the same.

I cannot claim to have exhaustively tested *every* aspect of this editor. What I gave it was a real-time, in use run through. I created programs with it, edited them, and even played around with them. Based on this level of experience with the system, I can definitely say that I like this editor.

When combined with MI-Prog's other program development tools, XEDIT becomes powerful. The other tools tried during this review were:

ASM/CMD - a Z80 assembler.
PACK/CMD - a program compression tool.
XDIR/CMD - an extended directory tool.

All performed well during the period. PACK actually makes the operation of XEDIT faster, since XEDIT is limited more by disk I/O than by its own editing ability. PACK takes out all unnecessary space, even comments if desired.

XDIR provides not just an extended directory listing for a disk, it also provides maps of disk storage and maps of the storage for an individual program. Each of these capabilities can add much to a programmer's ability if he is involved with direct disk work.

XEDIT, ASM, PACK and XDIR as well as other programs are available from Mi—Prog, PO Box 27014, Minneapolis, MN 55427. The prices are:

MODEL	I	II	III
XEDIT	$44.95	$89.95	$79.95
ASM	$34.95	$59.95	$49.95
PACK	$14.95	$19.95	$14.95
XDIR	$19.95		

These prices are for delivery on disk. It is possible to get Model I and Model III programs on cassette for less, but they are not cassette based programs.

T R Dettmann

notes

In our last issue we reviewed PROFILE II for the Model II. The comment was made that VERIFY DETECT OFF would improve performance. Another way to improve movement between modules is to eliminate the alternate directory. This is done during the formatting procedure under TRSDOS 2.0. In calling the format utility, type:

FORMAT {ALT=00}

This will improve performance in moving from the menu to various portions of the program.

Model II Users

In the NOTES column of the Nov/Dec 80 issue we talked about the dangers of an output of less than 25 to I/O port 255 (video controller) and the fix (a quick hand on the power switch). Please note that this happens only under machine language software control and not under Basic.

The problem occurs when the Video Processor IC is being programmed from user software. Originally, TRSDOS 1.1 and 1.1.2 would cause damage whenever you went to or from 40 characters per line. Versions 1.2 and 2.0 have fixed this problem in the Basic interpreter, but damage can still occur if you program in machine language and do not make use of the SVC calls.

The Model II will not operate successfuly with a Model III located next to the right side of it. The video flyback transformer of the Model III interferes with the disk operation of the Model II with both units on.

A note for Model I users moving up to the Model III, if you have USR routines in your programs, the places to POKE USR entry points are the same on both machines.

Do you have Model II Scripsit yet? If you do, be careful! When leaving the program, be sure to do it by exiting through the "SWAP DISKS OR EXIT" option under menu utilities. We don't have absolute proof yet, but it seems that not doing it may under some circumstances leave the disk file with your text open. When this happens, you may lose all of your text on that particular disk.

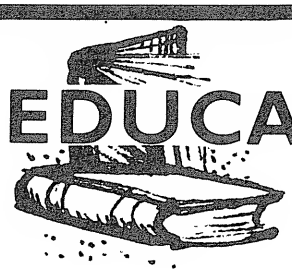
On the TRS-80 Model I, if you want to clear only part of the screen, you can have a machine language program load the HL register with the address of the beginning of the video memory (3C00H) plus the number location of the position to start clearing from (the same number you would use with a PRINT@). Then a CALL to 057CH will clear the rest of the screen.

Are you afraid of ARRAYs? No need to be. They can be very useful if you just learn to think in terms of where they can be a time saver. As a general rule of thumb, whenever you have to do an operation that is repetitive on some set of numbers, it is easier to put the numbers in an array and use a FOR..NEXT loop to do the calculation than it is to write the operation one statement after another. Another way to think of two dimensional arrays is as if they were tables of values. You can use the indices of the array to do a lookup in the table for what you want.

With TRSDOS 2.0 released for the Model II, please be aware that it is unwise to transfer Radio Shack programs written on TRSDOS 1.2. This is true of programs with machine language utilities only! However, be aware that even with multiple drives, the two systems will not talk to each other.

Another note about Radio Shack business software for the Model II is the inability to use a serial printer. The software is looking for certain printer information that can only be returned through the parallel port. A serial printer will cause program aborts in many places. Also, be aware that rewriting these programs to get rid of the problems will be a major undertaking.

Model III users, have you read your manual closely yet? Look at pages 12/24 and 12/25 in the Operation section, they have the addresses in ROM and RAM for some useful routines and codes. Page 12/22 in the Operation section tells you how to disable the BREAK key on the Model III (POKE 16396,175:POKE 16397,201). The whole of Chapter 12 gives you information on writing routines to interface with the ROM.



Med Systems!

Proven Educational Software

The Human Adventure allows movement through a human body's cardiovascular system. All major organ systems are accessible and fully described by the computer. A graphic CAT-scan constantly shows the user his position in the body. The exploration mode allows simple exploration, while the game mode places the user in a race against time to cure the patient of cancer using his knowledge of the body's layout. Recommended for reading age through adult.

The Playful Professor is a mathematics learning aid that provides tutoring in integer mathematics and fractions for the four basic operations. Demonstrated solutions are completed step-by-step in a blackboard format easily understood by grade school children. Problems are presented in a game format that places the pupil in a sixty room mansion. To win, the player must catch the ghost with the key, then get to the front door before the ghost (or other player) recaptures the key. Movement is based on problem solving. Difficulty may be different for each player, allowing parents to be beaten by their children. Recommended for age 4 through adult.

Money Master tutors the young child in the use of money. The child is allowed to wander freely by paying tolls or buying objects. The tutoring screen depicts money graphically, and interactively instructs in the use of coins. This includes making payments and receiving change. New mazes are generated for each game. Graphic obstacles are randomly chosen from a library of several dozen. An average game lasts 20-30 minutes. Recommended for early readers through adult.

Each program $9.95 on cassette for TRS-80 Level II 16K, or Model III 16K. All three on diskette - $29.95, Model I only.

Graphic 3-D Adventures

These machine language programs are the first in a new breed of adventure. Instead of wandering through the English language, typing GO EAST or GO WEST, you move through a collossal maze represented on the screen three-dimensionally. Hallways recede into infinity or come to dead-ends. Doors open to left and right. As you encounter objects, monsters, and mayhem, one or two word commands may be used. The command set is extensive and sophisticated. Movement is via the arrow keys. Graphics generation is instantaneous. Mazes are bit-coded and **HUGE.** There is simply nothing like these programs on the market today.

Deathmaze 5000 places you on the top floor of a five-story building. Each floor is a maze of twisting passageways. Floors are connected by elevators and open pits. You have but one goal. **Escape Alive!** Where is the only door out of this nightmare? Monsters, bats, mad dogs, hunger, and many more horrors plague your every step as you struggle to escape the most complex adventure ever written.

Labyrinth places you in a maze of gigantic proportions. But you are not alone! A minotaur searches for you, seeking a grisly meal. You must find weapons, spells, and treasures. You must deal with ghosts and cave gnomes. You *must* avoid the minotaur until the moment is right for the final battle. And if this isn't enough, the Labyrinth twists space and time so that you may not know whether you are coming or going!

Each program $12.95 on cassette for TRS-80 Level II 16K, or Model III 16K. Both on diskette - $29.95, Model I only.

ATTENTION DEATHMAZE FANATICS!

Still on the first level? You would look much better wearing the hat. But don't charge the wrong wall!

Satisfaction Guaranteed! All Med Systems Software products come with a 14-day moneyback guarantee. If for any reason you are not satisfied, return you order within 14 days for a prompt and cheerful refund.

Ordering Information. Orders are processed within two working days. Mastercard and Visa card holders please remember to include the expiration date. We pay all postage and handling within the U.S., Canada, and U.S. territories. European orders please include $2.00 for air post.

☐ Human Adventure	$ 9.95	$
☐ Playful Professor	$ 9.95	$
☐ Money Master	$ 9.95	$
☐ Deathmaze 5000	$12.95	$
☐ Labyrinth	$12.95	$
☐ Educational Diskette	$29.95	$
☐ Deathmaze-Labyrinth Diskette	$29.95	$
	TOTAL	$

Name

Street

City State Zip

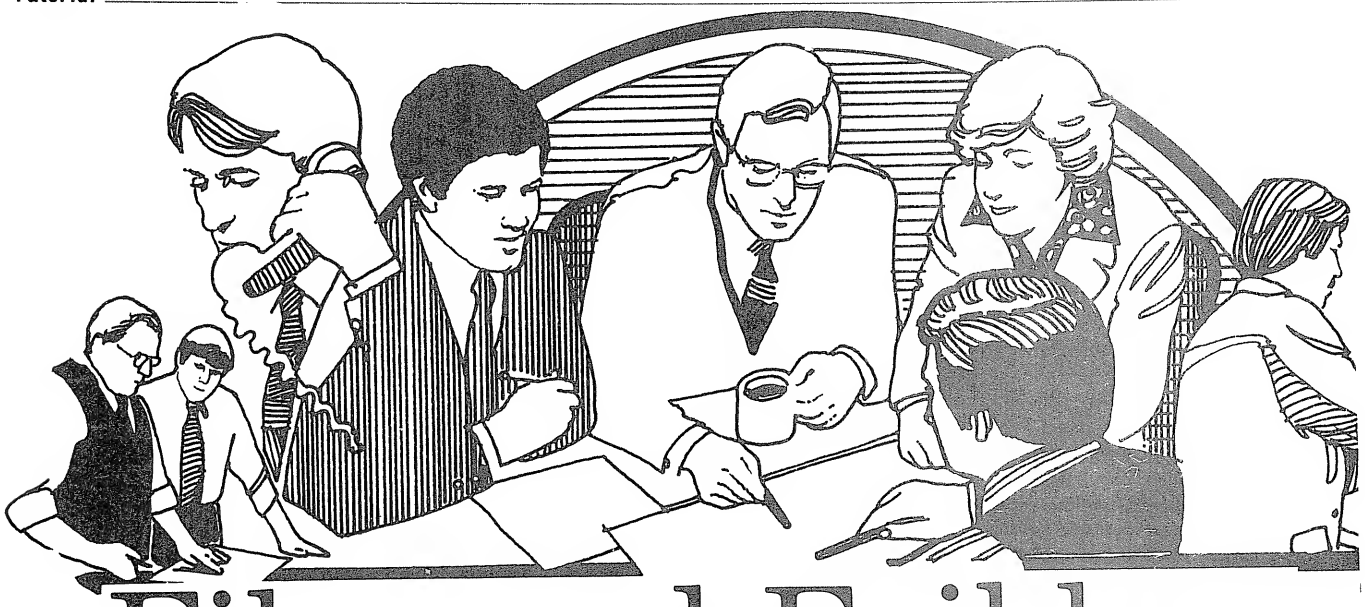
☐ MASTERCARD ☐ VISA ☐ Check

Mastercard or Visa #

Expiration Date

Med Systems Software

P.O. Box 2674 Department A58
Chapel Hill, North Carolina 27514
(919) 933-1990



Files and Foibles

Random Access Files

For Models I, II and III, with Disks.

T R Dettmann

Last issue, we went quickly over Sequential files because they are fundamental to our understanding of how files work. This issue, we'll go the next step and introduce Random Access files.

Why bother with Random Access files? That depends on your application. If your application is really sequential in nature or you have wildly varying data formats, Random Access files would be really difficult to work with. However, if you can specify the exact nature of your data, if you have to get to *any* piece of the data at *any* time without delay, then Random Access files are what you need.

A good example of where Random Access files can be used is a typical business inventory system. Generally, the businessman will want to look at the status of *any* item in the inventory at any time. If he wants to look at item 4965, he doesn't want to have to wait until the first 4955 items have been read before he finds his item.

With Random Access, you get item 4956 in the same time it would take to get item 1. Even on the TRS-80 Model I it seems instantaneous. Random Access files (on the Model II they are called Direct Access files) work differently than Sequential files. We'll have to learn some new programming techniques in order to handle them.

Working with Random Access

We will be interested in five new commands when working Random Access files. They are:

FIELD	Sets up the data record.
GET	Gets information from the file.;
PUT	Puts information in the file.
LSET	Puts data in a record.
RSET	Puts data in a record.

We also have to become acquainted with some new functions in Basic that are used to convert information to and from the format it is stored in on disk. These are:

To Disk	From Disk
MKI$	CVI
MKS$	CVS
MKD$	CVD

Before we go into any of these commands in detail, let's first look at how a random file actually works.

Disk space is set up as a series of "cubbyholes". Each little cubbyhole represents one "physical record" on the disk. That is, it is the size the system will handle in one chunk. This size is fixed at some number of bytes.

On Model I systems under TRSDOS, this number is fixed at 255 bytes, under other DOS's and on the Model II and III, this number may be some other size, but it is always fixed for a particular Random Access file.

Once the number is set for the size of the cubbyhole, we can get at any hole by its address. Let's take the simple example we mentioned earlier, an inventory. Let's assume that we want an inventory stored by part number with part numbers between 1 and 300. Even on a Model I, this can be stored as a Random Access file.

If we use the inventory number as the address of the item, we can get at any item directly with Random Access techniques. Let's see how that is done.

(Continued on page 106)

Goodies from GALACTIC

Specialty Programs for

TRS-80 Model I - II - III

EDAS 4.0 (Editor/Assembler)

This is the highly acclaimed *"USER ORIENTED"* Assembler for the TRS-80 Model II by GALACTIC. Loaded with features such as assemble to memory, block move, link to debugger, default filenaming, reverse video editing, warm start entry and much more. Now the programmer can write, assemble, test, and debug his code without ever leaving EDAS.

EDAS 4.0 with complete manual (120 pages)

Model II Version ..... Was $229.00

NOW ONLY $179.00

MASS/MAIL SYSTEM

This is the NAME and ADDRESS system for subscription control or large mailing lists. It will handle up to 10,500 records, with a worst access time of less than 15 seconds and usual access of less than one second. All adds, deletes, and edits are instant for the operator and are then completed later in a "batch monitor". Extensive documentation and ongoing support. Requires TRS-80 Model II and 2 disk drives minimum. Contact GALACTIC direct for detailed specifications and prices for your exact needs.

Model II Version ..... Contact GALACTIC for Price

STOCK MARKET MONITOR

This day to day market monitor is designed for the active trader. The system will track the performance of an issue against the market as well as against itself. The package comes with complete documentation and explainations of the formulas that are used by the program. The system is available for the Model I and the Model III TRS-80.

Model I and III cassette version $89.00

Model I and III disk version $99.00

INVENTORY MASTER

Tired of being a slave to an out-of-control inventory? Let GALACTIC'S INVENTORY MASTER put you in control of your inventory. INVENTORY MASTER operates on a TRS-80 Model I and Model III 48K disk system (Minimum of 2 drives with capabilities of up to 4 drives). Drive spanning capabilities allow you to track 2700 inventory items with a 4 drive system (5100 items for the Model III). Unique machine language sort allows for instantaneous item insertion (approx. 15 seconds with 2700 items in system). Item access can be immediate using system-supplied control numbers. Modeled after a proven main-frame system costing tens of thousands of dollars. Complete add/edit/delete capabilities supported. Placement of orders can be machine-generated as well as user-generated, with editing capabilities. Full report-generator included. Exquisitely documented.

Model I Version $159.00

Model III Version $259.00

MODEL II HOST I/O SYSTEM

From the original author of the TRS-80 HOST and TERM systems in the RADIO SHACK "COMMUNICATIONS PACKAGE". This system allows the full control of the HOST facility by your BASIC program. Set the number of nulls to be sent after a C/R, set a command line to be executed if carrier is lost, turn HOST on and off, switch to channel A or B as desired, enable and disable the ability for the remote terminal to "BREAK" BASIC, identify whether a character came from the HOST'S keyboard or from the REMOTE'S and more. No knowledge of assembler needed. All options may be accessed from BASIC or ASSEMBLER. Complete with detailed documentation. Don't isolate your Model II, Let outside terminals access it's computing power.

Model II with TRSDOS 1.2 ..... $179.00

Model II with TRSDOS 2.0 ..... $199.00

MAIL/FILE SYSTEM

This is the name, address, phone number data base manager that has set the standard by which other systems are compared. This system contains advanced editing and output capabilities. The TRS-80 Model I system will handle up to 600 records per file, while the Model III version will handle up to 1150 records and the Model II will handle 2500 records per file. All versions are file compatable and maintain constant sort indexes on both NAME and ZIP CODE. International PHONE numbers and ZIP CODES are supported. Thousands of code combinations are available. The Model II version also has a "word processor" type input editor and fast assembler sorting. Complete documentation is included with each version of MAIL/FILE.

Model I Version ..... $ 99.00

Model III Version $149.00

Model II Version $199.00

ULTRA TREK

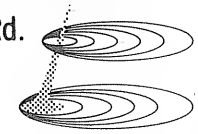
This is an all new concept for this type of game, and compares to the others like chess compares to checkers. ULTRA-TREK is a complex, logical game, intended for the serious contestant. It is doubtful that you will ever master this game, but you will certainly enjoy trying! This program requires a TRS-80 Level II, 16K or more. The program is written totally in BASIC and uses 15.5K of RAM.

Model I & Model III Version
(cassette only) $14.95

galactic software ltd.

A Division of GS & WS, Inc.

11520 N. Port Washington Rd.
Mequon, Wisconsin 53092
(414) 241-8030



Money Orders & COD's Shipped Within 24 Hours. Checks allow 2 weeks.

Circle 58

Random Access Commands

In order to understand the Random Access inventory, we have to understand each item in the inventory as a *record* in storage. A record is simply a group of logically related information. In this case, each piece of information about an inventory item will be stored together as a record.

Let's store everything as a series of string variables. We'll want to know the inventory number (it's address in storage), it's name, price, cost, amount on hand, who the supplier is, and the reorder level. We could store a lot more but this will do for the moment. Table 1 lists the expected size of string fields needed to store this information.

We have also included a rather large comment field that can hold any information of interest. For the moment, this will serve to illustrate how such a file is set up. We can make this whole scheme more efficient, on all TRS-80 Models, but in different ways. We'll worry about those details later.

With a total record size of 254 bytes for information, we'll assign one 256 byte sector to each record. In order to make this a file, *we* have to assign how the information is stored. That is the purpose of the FIELD statement.

FIELD

The field statement lays out the exact distribution of the items in each record. We have to specify where each item will go and how many spaces it will take up. For our file, we could write the FIELD statement like this:

FIELD #1,20 AS NM$, 8 AS PR$, 8 AS CS$, 4AS OH$, 20 AS SP$, 4 AS RL$, 190 AS CM$

This assigns variable names to fields in input/output buffer number 1 for each of the items in our inventory according to the field name assignments in Table 1.

The statement first identifies *which* file buffer it's talking

```
 10 REM**************************************************************
 20 REM
 30 REM            RANDOM ACCESS FILE DEMONSTRATION
 40 REM            TERRY R. DETTMANN        FILENAME: RANDOM/BAS
 50 REM
 55 REM            NOTE:  THIS PROGRAM IS WRITTEN SO THAT THE ONLY
 56 REM            STATEMENTS THAT YOU NEED TO TYPE IN ARE NUMBERED
 57 REM            BY TEN'S. ALL THE REMARKS EXCEPT THE HEADER ARE
 58 REM            NUMBERED IN BETWEEN. SIMPLY GO TO AUTO AND TYPE
 59 REM            THE STATEMENTS FOR THE NUMBER ON YOUR SCREEN
 60 REM**************************************************************
 65 REM            CLEAR SOME STRING SPACE FOR THE TEST
 70 CLEAR 2*MEM/3
 75 REM            SET UP ERROR PROCESSING FOR THE ERROR TEST
 80 ON ERROR GOTO 430
 85 REM            A$ WILL HOLD THE INPUTS IN MEMORY, N$ HOLDS THE
 86 REM            NAMES OF THE FIELDS
 90 DIM A$(10,7),N$(7)
 95 REM            OPEN THE FILE FOR RANDOM ACCESS, THIS MEANS THAT
 96 REM            WE CAN READ OR WRITE ON THE FILE
100 OPEN"R",1,"TEST/DAT"
105 REM            THIS IS THE FIELD STATEMENT FROM THE ARTICLE.
106 REM            EVERYTHING IS STORED IN STRING FORM. TRY REDOING THIS
107 REM            FOR THE CONVERSION FUNCTIONS FOR NUMBERS.
110 FIELD #1, 20 AS NM$, 8 AS PR$, 8 AS CS$, 4 AS OH$, 20 AS SP$,
 4 AS RL$, 190 AS CM$
115 REM            READ IN THE FILED NAMES FROM THE DATA STATEMENT
```

```
120 FOR I = 1 TO 7
130    READ N$(I)
140 NEXTI
150 DATA NAME,PRICE,COST,ON HAND,SUPPLIER,REORDER LEVEL,COMMENT
155 REM          LOOP TO ENTER DATA INTO THE FILE IN RECORDS 1-10
160 FOR I = 1 TO 10
170    CLS:PRINT:PRINT:PRINTTAB(10)"RANDOM FILE TEST":PRINT
180    PRINTTAB(10)"ITEM NUMBER: ";I
185 REM                    PROMPT FOR ITEM BY NAME THEN INPUT IT
190    FOR J = 1 TO 7
200       PRINT TAB(10) N$(J);
210       INPUT A$(I,J)
220    NEXT J
225 REM          PUT THE DATA INTO THE BUFFER FIELDS WITH LSET. ALSO
226 REM          TRY RUNNING THIS PROGRAM USING RSET INSTEAD OF LSET
227 REM          TO SEE WHAT HAPPENS.
228 REM          REMEMBER TO USE THE "M" CONVERSION FUNCTIONS IS YOU
229 REM          REDO THIS WITH NUMBERS INSTEAD OF STRINGS
230    LSET NM$=A$(I,1):LSET PR$=A$(I,2):LSET CS$=A$(I,3):LSET OH$=A$(I,4)
240    LSET SP$=A$(I,5):LSET RL$=A$(I,6):LSET CM$=A$(I,7)
245 REM          PUT THE BUFFER ONTO THE DISK AT RECORD LOCATION "I"
250    PUT 1,I
260 NEXT I
265 REM          LOOK AT RECORDS ON REQUEST AND COMPARE THEM
266 REM          WITH WHAT WE ACTUALLY INPUT FROM THE KEYBOARD
270 CLS:PRINTTAB(10)"RANDOM FILE TEST":PRINT
280 PRINTTAB(10)"ENTER A RECORD NUMBER (<=0 TO END)";
285 REM          PROVIDE FOR THE END OF THE ROUTINE WITH THE <=0 CHECK
290 INPUT N:IF N<=0 THEN 410
295 REM          GET THE REQUESTED RECORD. NOTE: IF THE RECORD NUMBER
296 REM          REQUESTED IS >10, THE PROGRAM WILL TAKE THE ERROR BRANCH
300 GET 1,N
305 REM          DISPLAY THE FIELD NAMES AND WHAT WAS PUT THERE NEXT TO
306 REM          WHAT'S ON DISK
310 PRINT,"ENTERED","IN FILE"
315 REM          REMEMBER: IF YOU ARE EXPERIMENTING WITH THE TYPE
316 REM          CONVERSION FUNCTIONS, USE THE "C" CONVERSIONS ON THE
317 REM          FIELD VARIABLES IN ORDER TO GET PRINTABLE NUMBERS
320 PRINT N$(1),A$(N,1),NM$
330 PRINT N$(2),A$(N,2),PR$
340 PRINT N$(3),A$(N,3),CS$
350 PRINT N$(4),A$(N,4),OH$
360 PRINT N$(5),A$(N,5),SP$
370 PRINT N$(6),A$(N,6),RL$
380 PRINT N$(7),A$(N,7),CM$
385 REM          WAIT NOW UNTIL YOU'VE HAD A CHANCE TO SEE THAT IT'S RIGHT
390 PRINT"PRESS ANY KEY TO CONTINUE";
395 REM          IF ANY KEY IS PRESSED, INKEY$ WILL HAVE A VALUE.
396 REM          IN THAT CASE, GO GET ANOTHER RECORD NUMBER
400 IF INKEY$<>"" THEN 270 ELSE 400
405 REM          END OF THE PROGRAM, ALWAYS CLOSE THE FILE BEFORE ENDING
410 CLOSE
420 END
425 REM          ERROR TRAP ... IT'S A GOOD IDEA TO CLOSE THE FILE HERE
426 REM          TO BE SURE NOTHING GOES WRONG
430 CLOSE
435 REM          ERR IS THE ERROR NUMBER ON MODEL II, ON MODEL I OR III
436 REM          SUBSTITUTE ERR/2+1 FOR THE ERROR NUMBER
440 PRINT"ERROR NUMBER: ";ERR;" ERROR LINE: ";ERL
450 PRINT"PRESS ANY KEY TO RESTART, <ENTER> TO END";
455 REM          THIS TIME WE NEED TO REMEMBER WHAT KEY WAS PRESSED
460 C$=INKEY$:IF C$=""THEN 460
465 REM          IF THE KEY WAS <ENTER> (ASCII VALUE 13), THEN WE'RE
466 REM          DONE, OTHERWISE, START OVER AT THE BEGINNING
470 IF ASC(C$)=13 THEN RESUME 410 ELSE RESUME 70
```

about (#1), and then assigns fields to each variable. Twenty bytes to NM$ for the item name, 8 bytes to PR$ for the item price, and so forth.

After a file is OPENed for Random Access, which we do simply by using an "R" where we would have used an "O" or an "I" in a sequential OPEN statement, then we FIELD it for our file.

Putting Information in the File

Once the file buffer is FIELDed, we have to put information into the buffer. This is done with the LSET and RSET instructions. LSET and RSET assign information to the field variables that we assigned with the FIELD statement. LSET puts the information in *left justified* form and RSET puts it in *right justified* form. It is important to remember that if the string to be put in the field is too long, then LSET and RSET will drop extra characters from the *right* of the string. If you did the following:

```
LSET PR$ = "Terry Dettmann"
PRINT PR$
```

or

```
RSET PR$ = "Terry Dettmann"
PRINT PR$
```

in either case you would get:

```
Terry De
```

printed out.

Now that we have it in the buffer, we have to put the information on the disk. We do this with the PUT instruction.

PUT/GET

PUT is one of two commands that we will use to actually move things to and from the disk file. PUT takes *anything* in the file buffer and places it in the designated record in the file.

For example, PUT 1,1 will put the current file buffer into record number 1. PUT 1,IN will put the file buffer into the record specified by the variable IN. The disk operating system can go directly to that record in the file and it does so when putting the record on disk.

To retrieve the record for use, we use the GET instruction the same way we used PUT. We specify the number of the file and the number of the record to pull from the disk and put in the buffer. GET 1,1 will get record number 1 from file number 1. GET 1,IN will get record number IN from file number 1.

When we GET a record, we don't change a thing on the disk. We just make a copy of it in the memory buffer that we can use to fill our program variables with. Once we have a record in memory, then we use the FIELD variables to assign the values to our program variables.

After we GET a record, we can PRINT the values of the field variables, and even use them to write assignment statements. We cannot use them on the left side of an equal sign however. With our previous FIELD statement, if we ever in the program wrote:

```
NM$ = "Terry Dettmann"
```

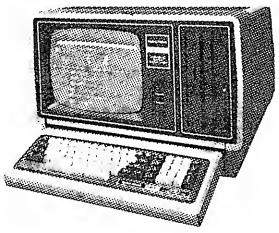
without LSET or RSET, then NM$ would become a normal string variable and no longer a FIELD variable. However, we could use it like this:

```
NAME$(I) = NM$
```

In this case, we are assigning the value of the variable NAME$(I) to be whatever is in the FIELD NM$. In this way we can transfer information out of the buffer for use.

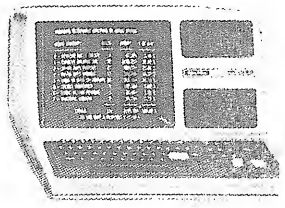
(Continued on Page 110)

MODEL II



$ DISCOUNT $
TRS-80® DEALER A301

MODEL III



COMPUTER SPECIALISTS

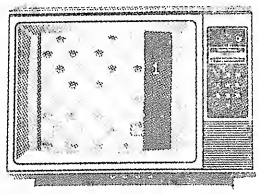
26-4002
64K 1 Drive
$3440.00

26-4160 1 Drive EXP.	$1035.00
26-4161 2 Drive EXP.	1575.00
26-4162 3 Drive EXP.	2115.00
26-4501 Gen. Ledger	180.00
26-4502 Inventory	180.00
26-4503 Payroll	360.00
26-4554 Acct. Rec	180.00
26-4701 Fortran	270.00
26-1157A Daisy Wheel	2290.00
26-1158 Daisy Wheel II	1799.00

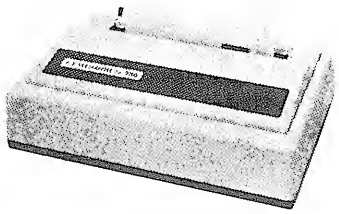
26-1155 Quick Printer II	$187.00
26-1145 RS-232 Board	84.00
26-1140 "O" K Interface	249.00
26-1141 "16" K Interface	359.00
26-1142 "32" K Interface	469.00
26-1160 Mini Disk - Drive O	419.00
26-1161 Mini Disk - Additional	419.00
26-1154 Lineprinter II	699.00
26-1165 Line Printer V	1710.00
26-1159 Lineprinter IV	859.00
26-1166 Line Printer VI	1080.00
26-1563 Scripsit - Disk	79.00
26-1566 Visicalc	83.00
26-1562 Profile	72.00

26-1061 4K I	$630.00
26-1062 16K III	888.00
26-1063 32K III 2-Drives, RS232	2225.00

COLOR



26-3001 4K	$360.00
26-3002 16K	540.00
26-3010 Color Video	360.00
26-1206 Recorder	54.00
26-3008 Joysticks	22.50



ALL OTHER R.S. SOFTWARE FURNITURE, STANDS, CABLES AND ACCESSORIES AT DISCOUNT FROM CATALOG PRICE.

Novation Cat Modem . . $149.00
CCA Data Management System 72.00
Adventure Games Games 1-9 each 14.00

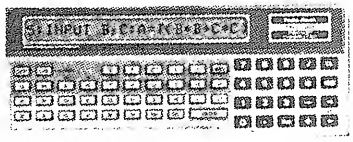
Acorn Software Products, Inc.

CENTRONICS

Fast 100 CPS Centronics
730 Printer 577.00
Text Quality Centronics
737 Printer 737.00

Model II Cobol Compiler $360.00
Cobol Run Time Package $36.00

Pocket Computer



26-3501 1.9K P.C.	$225.00
26-3503 Cassette I/F	45.00
14-812 Recorder	72.00

GAMES:
Alien Invasion $9.00
Stock Market 9.00
Star Trek 9.00
Block Em 9.00
Ting-Tong 9.00
UTILITIES:
System Savers 14.00
EDUCATION:
Language Teacher 18.00

FREE: COMPUTER CATALOG UPON REQUEST

1-800-841-0860 Toll Free Order Entry

MICRO MANAGEMENT SYSTEMS, INC.

No Taxes on Out Of State Shipments

Immediate Shipment From Stock on Most Items

DOWNTOWN PLAZA SHOPPING CENTER
115 C SECOND AVE. S.W.
CAIRO, GEORGIA 31728
(912) 377-7120 Ga. Phone No.

R.S. 90 Day Limited Warranty
F-48 Form Provided

Largest Inventory In the S.E. U.S.A.

*TRS-80 is a registered trademark of the Tandy Corp.

The program listing with this installment gives you a short Random Access file with the structure we've outlined here to play with. It allows you to enter 10 items into a file, and then read them back one by one and compare them to what you had in the file to start with. Try it for a start.

What About End Of File?

In a Random Access file, the system takes care of the end of file automatically. If you PUT to a record number higher than the current end of the file, the system assigns enough space to handle all the record numbers up through the record number you asked for.

For example, let's say that you had ten items in the file with record numbers 1 through 10. If you next add a record number 20, the system will allocate disk space for records 11 through 19 and then put in record 20 as the new end of file. These spaces will be there *even though nothing is stored in the records*.

If you were to GET record 11 say, and try to use it, you would get whatever happened to be on the disk at that point before it was allocated to your file. Generally, it's a mess of garbage!

To find out what the highest record number in a Random Access file is, the function LOF(N) is provided in Basic. It returns the highest record number of file number N. It really doesn't matter whether the file is Sequential or Random, LOF works just the same, but the number is really only meaningful for most purposes with Random files.

The function of EOF(N) doesn't work at all with Random Access files. It gives you no indication of whether the end of file has been reached with Random Access files. If you try to GET a record past the end of the file, all that will happen is that you will have your program stop on a Basic error. Try it with the sample program by trying to GET record 100.

Numbers

What do we do about numbers in a Random Access file? We have already stored them as strings. However, this can be wasteful of space. If we want to store a number like 30000 as a string we need 5 bytes, but as an integer it only takes 2 bytes in memory. Can we save it in only 2 bytes on the file?

The answer of course, is yes. But, we have to be tricky about it. The reason is that everything in the file has to be stored in the file as if it were ASCII coded. This is fine for strings, but not for numbers. Unless we want to convert our numbers to strings for storage, we have to do something else.

Basic gives us three functions to convert numbers to equivalent length strings and three more to convert them back again. They are listed in Table 2. Notice that in going from number to string, they always start with "M" and have a "$" to indicate that they give string values. To go the other way, they always start with "C".

Now, the result of the number-to-string functions does not give us a printable number. For integers, what we have is a 2 byte, binary number that represents our integer. In fact, you couldn't tell it from the number stored in memory. What has happened is that the system has been fooled into *thinking* this is a string.

Each variable has assigned to it a variable type code. That is how Basic knows what type of variable you are looking for when you use it. The number-to-string functions simply change this code! This makes it compatible with the I/O routines which think they are handling strings.

(Continued on Page 112)

HOMES for the TRS-80

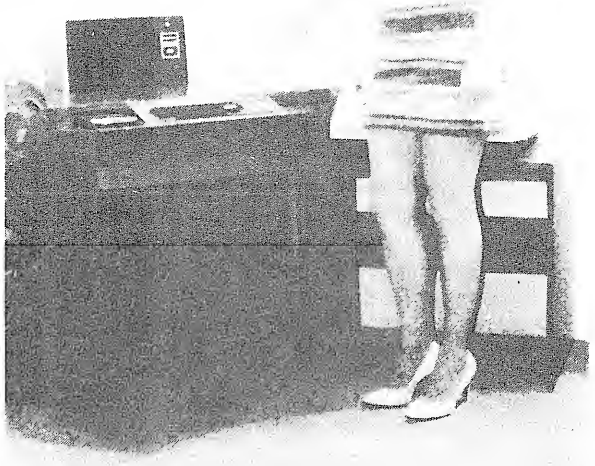
Featuring

High Quality, Commercial, and Economy Models

Computer Consoles from $129.95
Printer Stands from $39.95

Complete TRS-80 Business System, built-in

Special Educator Series



Custom furniture for the TRS-80 office or home decor.

AVS is a manufacturer, dealer, and consultant who specializes in products for the TRS-80 and their uses. You will note we do not handle multiple brands of each type of product. When we decide to handle a product, we try to analyze all the manufactures of that product. Then we select the manufacture we feel provides the greatest features, performance, and reliability relative to the cost and needs for the TRS-80. Though many times there may be more than one manufacture that meets our requirements, we prefer to only handle the one, and give our customers maximum support in the use of that product relative to the TRS-80.

AVS Custom Computer Furniture: Home for the TRS-80 takes the orphan modules of the TRS-80 and turns them into one homogenious unit. Our economy, commercial, and high quality consoles all house the keyboard, interface, cassette, and monitor. In addition, we have a special series for the schools. Our lines consist of computer consoles, line printer stands, storage hutches, carrols, and booths.

VISTA Disk Drives: The Vista drives are preferred by many system specialists, including ourselves, for their speed and reliability. They are available in both 40 and 77 tracks, providing memory in excess of 197K per disk, In addition, a module is available that allows the TRS-80 to run the Vista drives at double density, thus, providing more than 394K of memory per disk.

EXATRON Stringy Floppy: The stringy floppy is the ideal mass storage system for most home, school, and some small business systems, because of its simplicity of operation, high reliability, low cost, and speed. Unlike disk, it does not require the expansion interface nor the 11K of computer memory, to perform its many similar functions.

BASE II Line Printer: This printer performs many functions not even availabe in printers three times its price. Besides having a vertical density of 144 half dots to the inch, you can program your own character fonts. In addition, we have a special modification that allows the printer to run without an expansion interface.

AVS Green Thing: Our green screen works on both models I and II of the TRS-80. It not only performs the same tasks, but costs one-half to one-third of its competition.

Miscellaneous Accessories: For our customer convenience, we provide a series of AC outlet strips, line filters, and cooling fans.

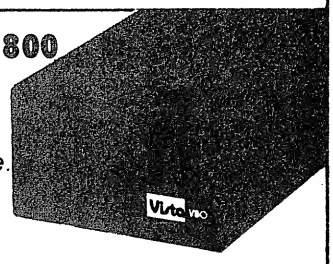
Circle 65

AVS AUDIO-VIDEO SYSTEMS

2485 AUTUMNVALE AVE.
SAN JOSE, CA. 95132
PH. 408-946-1265

PRODUCTS for the TRS-80

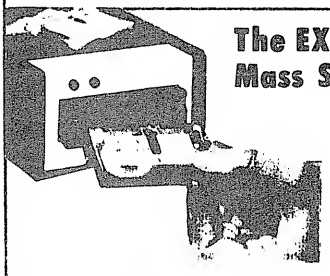
The VISTA V-80 & V-800 Disk Drive System



- *40 & 77 Track Units*
- *12 msec Access Time.*
- *High Reliability.*
- *Double Density.*
- *394K per Diskette.*

List $395.00

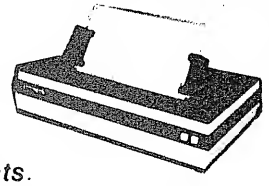
The EXATRON Stringy Floppy Mass Storage System



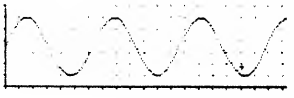
- *Access 15 x Cassette.*
- *Simplicity of Operation.*
- *Direct Keyboard Cont.*
- *Uses only 1K Memory.*
- *60K per Waffer.*

List $249.50

The BASE II Graphic Line Printer

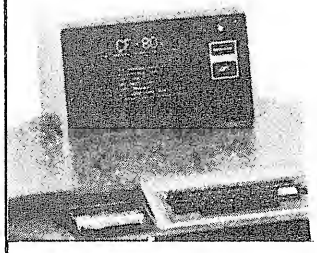


- *Tractor/Friction Feed.*
- *5 Charactor Densities.*
- *Programable Aux. Fonts.*
- *2K Buffer*
- *72V x 99H Dot Res.*



List $699.00

The AVS Green Screen Thing SC-80

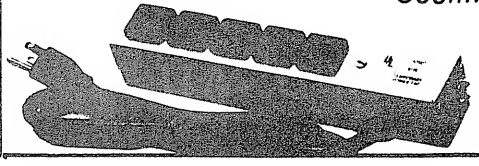


- *Improved Contrast.*
- *Reduces Eye Fatigue.*
- *Enhances Legibilty.*
- *Decreases Glare.*
- *System Looks Prof*

List $7.95
(Add $1.00 for shipping & handling, and California residents add 6% sales tax.

Miscellaneous System ACCESSORIES

- *AC 3 Wire, Multi-Outlets.*
- *AC Line Filters*
- *Cooling Fans*



Try printing this:

PRINT 100;MKI$(100)

Not very illuminating, is it? But if you convert it back again, it is useful:

PRINT 100;MKI$(100);CVI(MKI$(100))

It is a small price to pay to save some space. Table 2 gives the number of spaces needed in the FIELD statement for each type of variable.

If we redo the FIELD statement for our hypothetical inventory to let price and cost be single precision and the other numbers to be integers, then we can use the FIELD:

FIELD #1, 20 AS NM$, 4 AS PR$, 4 AS CS$, 2 AS OH$, 20 AS SP$, 2 AS RL$, 198 AS CM$

We have saved 8 spaces for the numbers and added them to the comment field. Try it with the sample program, but remember to use the conversions to convert the numbers to strings and back again.

Where do we go from here?

Next issue, we will go more into Random Access techniques and introduce more advanced methods of handling these files. We will show applications of the conversion functions, some methods to make your FIELDing easier, and advanced ways of handling files when the indexing number is not a nice neat integer between the limits of 1 and 300. What do you do when you want to do it with numbers between 1 and 1000 or 1 and 10000? How about if everything is to be indexed by name, or Zip code?

We'll talk about these methods and more next issue, when we continue with Random Access file techniques.

Table 1
Inventory Fields

Item	Size	Field Name
Name	20	NM$
Price	8	PR$
Cost	8	CS$
On Hand	4	OH$
Supplier	20	SP$
Re-order Level	4	RL$
Total	64	
Comment	190	
Field Size	254 spaces	

Note: These fields are set up for illustration only and do not make use of the space as efficiently as possible. That will come later when the techniques are known well enough to be second nature.

Table 2
Format Conversion Functions

Number to String

Function	Variable type	Field Size
MKI$	Integers	2
MKS$	Single Precision	4
MKD$	Double Precision	8

String to Number

Function	Variable type	Field Size
CVI	Integers	2
CVS	Single Precision	4
CVD	Double Precision	8

In a desperate race against the sun you search for SMAEGOR Monarch of Dragonfolk, who has kidnapped the Princess of the Realm and holds her in a distant and unknown place. In a quest for Honor and glory, you must search the land, seeking out the tools needed for the ultimate confrontation. On The River Delta, in the abandoned Temple of Baathteski, Goddess of the Blade, everywhere, clues abound. But WHERE is the Princess?

Now, as never before, the genius of CHARLES FORSYTHE shines in this new machine language ADVENTURE. DRAGONQUEST! Can YOU save M'lady from the iron clutches of SMAEGOR?

TAPE $15.95 *Dealer Inquiries Invited* **DISK $21.95**

Circle 68

THE PROGRAMMER'S GUILD
BOX 66, PETERBOROUGH, N.H. 03458
(603) 924-6065 AFTER 6 PM EST
MASTERCHARGE AND VISA ACCEPTED

Echo your printer to your screen

Bob Foster
Merriam, KS

For the Model I TRS-80

Did you ever wish you could see on the video screen what was being printed on the line printer? Well, you can....

By adding one Basic line to some of your programs, what goes to the line printer goes to the screen as well. This is especially handy when running a long print job. With some programs you can actually proof and plan future input while the printer is still working. It's much easier than waiting for the printer to finish.

Figure 1 shows the expanded and amplified program that condenses to one line as shown in Figure 2. To use the echo program, just add the code in Figure 2 to your program. Place the first line of this program right after the line in your program that clears string space. Place the second line where you are just through with the echo program. Before you go to another Basic program, remember you must make the POKEs in line 9000, reset, reboot or power off.

The program works by placing machine language code into a dummy string within the Basic program. You do not need to reserve memory, since the machine code actually ends up within the Basic program. The POKEs intercept a character on its way to the line printer and send it to the echo program. The echo program then sends the character to the video screen and on to the line printer.

Line 140, Figure 1, has a dummy string to hold the code. Line 160 locates the dummy string. The address of this dummy string (lines 180-200) is POKEd into the line printer control block. To POKE the data into the dummy string, line 240 converts the address to one decimal number. Lines 260-280 read and POKE the data into the dummy string.

The values in the data statement are decimal equivalents of these hexadecimal numbers: 79, CD, 3A, 3, C3, 8D, 5. In assembly language these hexadecimal numbers represent:

```
79          LD      A,C     ;THE CHARACTER ABOUT TO
                            ;BE PRINTED IS IN REG C.
                            ;WE ALSO WANT IT IN A.
CD3A03      CALL    033AH   ;OUTPUTS THE CHARACTER
                            ;IN REG A TO THE VIDEO.
C38D05      JP      058DH   ;OUTPUTS THE CHARACTER
                            ;TO THE LINE PRINTER
                            ;AND RETURNS FOR THE
                            ;NEXT CHARACTER OR TO
                          ;CONTINUE THE BASIC
                          ;PROGRAM.
```

SAVE YOUR TIME ...AND YOUR MONEY.

Instant Software has the two best mail list programs available for your TRS-80 Model I and Model II.

Mail/List for Model I and Model II

This mailing list program maintains separate alphabetical and zip code files in *constant sort*. When you add a name to your list, it will *automatically* be inserted into its correct position in the files; therefore it's always ready to print labels!

It will record your information in these fields: NAME, ADDRESS, CITY, STATE, ZIP, PHONE NUMBER, PHONE EXTENSION and a five character CODE field. You have the choice of a 3 line, 4 line, or user defined label format. It can even include (optionally) a message line on your label!

The programs most outstanding feature is its sorting capabilities. Mail/List allows you to choose which names you want to be printed from the whole list. For example, all people in one zip code, or all people named Jones, who are living in a particular city or state. For any name in your list you can assign a code within the CODE field. You can then specify the code when printing labels, and only names with that code will be printed out. You can specify up to 9 different codes!

Every business and organization will save time and money with Mail/list to keep track of customers or members.

TRS-80 Model I version, Order No. 5000RD $99.00.
Requires 16K RAM, Expansion Interface with at least 16K RAM, one disk-drive and a printer.

TRS-80 Model II version, Order No.5001RD $199.00.
Requires 64K RAM and printer.

WRITE FOR OUR NEW CATALOG

One-D Mailing List

Here is a mailing list system that can be run on only ONE disk-drive! You can have up to 17 fields of selection for name/address retrieval. •Disk versatility allows you to add, delete, or change the numerous details stored in the system.

•Features of the One-D Mailing List includes: •Automatic name sort (alphabetically or by ZIP code). •Rapid access to any name on file. •Easy error correction and recovery. •Prints selective name listings. •Revise or update listings at any time. •Up to 2500 names on-line (with 4 drives). •Prints a list of all names on file. •Prints mailing labels. This package requires the following minimum system: 1. A TRS-80 Model I Level II microcomputer with 16K RAM. 2. An Expansion interface with 0 to 32K of RAM. 3. A single disk drive (extra drives optional). 4. A printer. 5. Any TRSDOS compatible Disk Operating System.

Order No. 0123RD $24.95.

TO ORDER: See Your Local Instant Software Dealer or Call Toll-Free 1-800-258-5473.

Instant Software™

PETERBOROUGH, N.H. 03458

SOME STRAIGHT TALK ABOUT DISK DRIVES

- DON'T BE CONFUSED BY ALL THE BRAND NAMES YOU SEE IN THE MARKET PLACE, THERE ARE VERY FEW MANUFACTURERS OF THE BASIC DRIVE CHASSIS, ALL THE OTHER NAMES ARE THOSE OF THE ASSEMBLERS OR THE RETAILERS.
- AS MANUFACTURED, THE DRIVE WILL NOT RUN ON A TRS-80$, IT MUST BE MODIFIED BY THE ASSEMBLER.
- THE QUALITY OF THE DRIVE DELIVERED TO YOU IS DEPENDENT ON BOTH THE MANUFACTURER AND THE ASSEMBLER, THE BEST CAN TURN TO JUNK IF THE ASSEMBLY IS IMPROPERLY DONE.
- THE POWER SUPPLY AND CASE ARE VERY IMPORTANT COMPONENTS OF THE COMPLETE DRIVE, THE CASE MUST ALLOW PROPER COOLING AIR FLOW, AND THE POWER SUPPLY MUST MAINTAIN TWO CONSTANT VOLTAGES.
- YOU MUST DEPEND ON THE COMPANY SELLING YOU THE DRIVE TO SERVICE IT AT REASONABLE COST WHEN IT FAILS YOU, THE MANUFACTURER IS NOT EQUIPPED TO DO THIS!
- THE BEST MEASURE OF QUALITY IN A DRIVE IS IT'S SPECIFICATIONS, WILL IT HANDLE DOUBLE DENSITY, WHAT IS THE TRACK TO TRACK ACCESS TIME, THE ANSWERS TO THESE TWO QUESTIONS INDICATE THE PRECISION OF IT'S COMPONENTS.
- WHAT KIND OF DRIVE SHOULD YOU BUY ? LEVEL IV HAS CHOSEN TO DISTRIBUTE EXCLUSIVELY, THE MPI LINE, ALL MODELS OF MPI ARE DOUBLE DENSITY RATED AND REQUIRE ONLY A FIVE MILLI-SECOND TRACK TO TRACK ACCESS TIME.
- WHAT DO ALL THE MODEL NUMBERS MEAN?
  B-51= 40 TRACKS SINGLE HEAD SINGLE SIDE B-52= 40/40 TRACKS DOUBLE HEAD DOUBLE SIDE
  B-91= 80 TRACKS SINGLE HEAD SINGLE SIDE B-92= 80/80 TRACKS DOUBLE HEAD DOUBLE SIDE
  (DOUBLE HEADS)-READ BOTH SIDES OF DISK (DUALS)-TWO DRIVES IN ONE CASE (RAW)-NO POWER SUPPLY OR CASE
- WHERE SHOULD YOU BUY YOUR DRIVE, LEVEL IV IS ONE OF THE OLDEST AND LARGEST DISTRIBUTORS OF TRS-80$ EQUIPMENT, LOOK AT THE ADS IN YOUR OLD MAGAZINES, MANY OF THE ADVERTISERS ARE NO LONGER IN BUSINESS, LEVEL IV HAS BEEN A LEADER SINCE THE BEGINNING, WE STAND BEHIND OUR PRODUCTS, AND WE'LL BE HERE WHEN YOU NEED HELP.
- WHERE DO THE NATIONALLY KNOWN AUTHORS BUY THEIR DRIVES ? LEVEL IV CAN SHOW COPIES OF SALES RECEIPTS FOR DRIVES TO MOST OF THEM, LANCE MICKLUS, SCOTT ADAMS, VERNON HESTER, BOB (CAPTAIN 80) LIDDIL, DICK BALCOM, KIM WATT, ETC. TO THESE MEN COMPUTING IS NOT A HOBBY, IT IS THEIR PROFESSION.
- CALL FOR OUR LOW PRICES ON NEW AND USED DRIVES, REMEMBER, WE ALSO TAKE TRADES, AND PAY THE SHIPPING!

Circle 70

LEVEL IV PRODUCTS INC. 32238 SCHOOLCRAFT, LIVONIA, MI 48150
PHONES: MI (313) 525-6200 OTHERS 800-521-3305 (TOLL FREE)

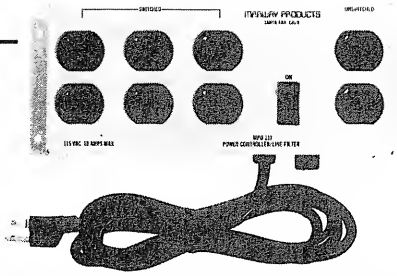
$ a tradeark of the RADIO SHACK DIV. of TANDY CORP.

```
100 REM  ECHO L/P TO THE SCREEN
110 REM
120 REM     BY ROBERT D FOSTER
130 REM
140 EC$="XXXXXXX" : REM DUMMY
    STRING THAT WILL CONTAIN THE
    CODE
150 REM
160 EC=VARPTR(EC$) : REM LOCATES
    ADDRESS OF THE CODE (EC$)
170 REM
180 EL=PEEK(EC+1) : REM LEAST
    SIGNIFICANT BYTE OF CODE
    ADDRESS
190 REM
200 EM=PEEK(EC+2) : REM MOST
    SIGNIFICANT BYTE OF CODE
    ADDRESS
210 REM
220 POKE 16422,EL : REM ADDRESS
    OF CODE INTO THE L/P CONTROL
    BLOCK
230 POKE 16423,EM
240 EC=EL+256*EM : REM ADDRESS TO
    DECIMAL TO USE IN POKE STATE-
    MENT
250 REM
260 FOR EI = 0 TO 6
270   READ ER : REM READ DATA
      (MACHINE CODE)
280   POKE EC+EI,ER : NEXT : REM
      POKE CODE INTO DUMMY STRING
290 REM
300 DATA 121,205,58,3,195,141,5 :
    REM MACH LANG CODE IN DECIMAL
310 REM
9000 STOP
9010 POKE 16422,141 :
     POKE 16423,5 : REM RESTORE
     THE PRINTER TO NO ECHO
```

FIGURE 1
ECHO PROGRAM WITH REM STATEMENTS

```
100 EC$="XXXXXXX":EC=VARPTR(EC$):
POKE 16422,EL:POKE 16423,EM:EC=EL
+256*EM:FOREI=0TO6:READER:POKEEC+
EI,ER:NEXT:DATA121,205,58,3,195,
141,5
9000 REM RESTORE PRINTER TO NO
     ECHO
9010 POKE 16422,141:POKE 16423,5
```

FIGURE 2
COMPRESSED ECHO PROGRAM W/O REM STATEMENTS



NO "GLITCHES", SURGES OR INTERFERENCE

Clean power distribution for your:

- **Microcomputer system**
- **Audio Visual system**
- **Lab instrumentation**

Marway Products' new MPD 117 AC Power Controller upgrades a standard AC utility outlet into a convenient noise-free source of AC power. The MPD 117 features:

- High energy EMI filter
- Transient voltage suppressor
- 10 amp circuit breaker
- 6 switched outlets
- 2 direct (unswitched) outlets
- Illuminated "on/off" switch
- UL recognized components

At $89.00 (plus tax and shipping) the MPD is the low-cost solution to your power distribution needs, and the best dollar value available in AC power distribution and noise suppression products.

Circle 73

Marway can solve your power distribution problems and save you money.

MARWAY PRODUCTS INC.

2421 S Birch St , Santa Ana, CA 92707 (714) 549-0623

MODELS I & III UPGRADED

Check Our Prices!

- TRS-80 Model II Level II 16K **$925.00**
- 16K Memory upgrade for Mod I, III & Color **$59.95**
- New Archbald Speed up board **$44.95**
- 5¼" Verbatim Diskettes (Box of 10) **29.95**
- Vista V-50 40 track double sided disk drive **$395.00**
- Vista V-800 80 track double sided disk drive **$595.00**
- Vista V-8000 160 track double sided disk drive **$795.00**

Drives include case & power supply but no cable.

- **Cable for 2 drives $23.95**
- Printer Multiplexer Box, handles up to 4 printers at the same time, user selectable. ★
- **Epson MX-80** ★
- Vista V-300 Daisy Wheel Printer ★
- EPROM Programmer ★
- MAXI-ZAP for Model III **$79.95**

(Greatly enhanced Super-Zap like Utility for Model III, written in machine code. Reads Model I and Model III Diskettes!)

- Shugart Technology 5¼" Winchester 6.3 Megabyte Drive **$2395.00**
- 506 Controller Board **$1500.00**

CALL OR WRITE FOR FURTHER INFORMATION

★ Please call for these prices

VISA | master charge THE INTERBANK CARD

Circle 74

Micro Mainframes
714 Alhambra Blvd.
Sacramento, CA 95814
(916) 447-7048

Prepare Income Taxes — Yours, Too! On Your Microcomputer 40 INCOME TAX PROGRAMS

Most "tax preparers" charge $35 to $50 per hour — and up! Interesting seasonal spare-time work. Your computer figures taxes easily, and with printer, types them too. Our 130-page book guides you in programming income tax forms.

Do your own taxes, Do them for others, too. Here are more than 40 income tax, easy to follow programs, with variables.

One chapter discusses depreciation of microcomputers used for profit. Another gives programming tips for newcomers. There are programs that "look up" taxes from the "tax tables" and the "tax schedules". Others for earned income credit, installment sales, rental statements and many others — 40 plus in all.

Written in TRS-80 MOD. 1 basic (T. M. Reg. — Tandy Corp.) with command listings so other brand users can convert to their computer system.

Book* — "40 Income Tax Programs" — $16.95 ppd.

Available at many computer stores. Mo. res. add 5% sales tax.

GOOTH SOFTWARE

Dept. 221 931 So. Bemiston, St. Louis, Mo. 63105

MasterCard

Master Charge, Visa, Check or Money Order.

*This book may be tax deductible if it assists you in your taxes, or in tax work for profit.

VISA®

Circle 75

TRS-80 LEVEL II

- **PLANET LOCATOR** - As the name implies, this program will compute the location and display of the positions of the planets. It also provides important statistical data. This is a valuable tool for the ASTRONOMY BUFF (16K) ..... $ 9.95
- **LEAVE ONE** - This game places you against the computer in a match (taking turns and using game rules) to pick up objects and leave the last object for your opponent. The game itself can be played without a computer, but there is a trick to winning. (4K) ..... $ 6.95
- **CHECK STORAGE AND YEAR END SUMMARY** - This two-program system handles the capture, sorting, summarizing and various other important functions needed to maintain you data. Graphics have been incoporated to simplify keying in the checks. The second program is the year end summary report. This is very useful at tax time or just as a expenditure analysis report. (16K) $11.95
- **AMORTIZATION** - This handy little program will calculate a six-column display as follows:
  COL. 1 - Payment Number
  COL. 2 - Monthly Payment Amount
  COL. 3 - Portion Applied To Principal
  COL. 4 - Portion Applied To Interest
  COL. 5 - Remaining Principal Balance
  COL. 6 - Principal Paid Back To Date
  Totals at the end show total interest and total amount paid. (4K) ..... $ 6.95
- **MESSAGE CODER-DECODER** - Have any written material you wish to keep private. Select one of 250 coding methods which will scramble and change your message so that only the computer can decode it back to its original form. (4K) ..... $ 6.95
- **CONTRACT PROGRAMMING:** For as little as $20.00. Send a written summary of desired system. You will be contacted as to price.

Circle 76

HOW TO ORDER

Send Check or money order to **ARLINGTON MICRO-SOFT, INC.**
4800 Crestmont Court Arlington, TX 76017

Name _____
Street _____
City/State _____
Zip _____ Phone (___)_____

Texas Residents add 5% Sales Tax Allow 4 to 6 weeks for delivery

UNCLASSIFIED ADS

$2.50 per half inch and $2.50 for each additional half inch per insertion. Ads must be typewritten or printed clearly, and must be accompanied with payment in advance. No telephone orders or "bill me's" will be accepted. Unclassified ads are always set in the type on this page, with the first line in bold type. Use the ads on this page as a guide to figure your size and payment. Send to: 80-U.S. 3838 South Warner St., Tacoma, WA 98409

TRS-80 EQUIPMENT! DISCOUNTED TO YOU Centronics 737(same as Line Printer IV) $775. Okidata Microline 80 40/80/132 column plus graphics. Handles single sheet/roll/fanfold paper $504 (cable for either $25). 16K RAM set/8 $32, Mini Flippy Kits (punch and use both sides of your diskettes) $7.50. 100 =/- programs on disks $39. Novation Modems, acoustic $145, direct $159. Diskettes X10 BASF $27, 3M $32, Maxell $36, Dysan $42. All equipment factory warranted. Write for the complete list. Specify Model I or II. Richard Smith, 3078 Baronscourt Way, San Jose, CA 95132 (408) 258-5202(071)

WANTED: "SCRABBLE TYPE"CROSS- word game for the TRS-80, any model. Write to Margaret, PO Box 7112, Tacoma, WA 98407 (071)

DEALERS WANTED. GREAT DEMAND FOR school soft-ware with educators, students & families. AA Educational Services, 7211 South 180th St., Kent, WA 98031 (071)

ARTIFICIAL INTELLIGENCE STUDY proceedings of 2nd Seminar of Scientific Go Theory now available. Send $12.00 to Sabaki Go Company, PO Box 4195-U, Wilmington, DE 19807 (031)

CAN YOU DEMONSTRATE L2 16K TO schools, students, families in your area? Write for dealer discount schedule. AA Educational Services, 7211 S 180th, Kent, WA 98031 (071)

FOR SALE!! LYNX 300 BAUD ORIGINATE/ answer modem with good terminal program for sale. Works with 4K Level 1 and up. Not used much - needs a modular phone jack (this is a direct connect modem) which we don't have. Call Andy Gillcrist nights (503) 774-2580 (031)

MAKE YOUR LIST ILLEGIBLE CONCEAL your access codes. $5.00 cash. Make the bums look like bums. AA Educational Services, 7211 South 180th, Kent, WA 98031 (071)

USED PRINTERS: CENTRONICS 730 LINE printer. Used very little since purchase...$440.00 Radio Shack Line Printer I with tractors, excellent condition...$600.00 JWC Corporation, Box 627, Union, OR 97883 (503) 562-5137 Call now! While the supply lasts! (031)

5 FOR 1 PROGRAM EXCHANGE, SEND 1 unreserved program that runs plus $5.00. Receive 5 programs in exchange. 5 for 1 Program Exchange. 25824 124th S E, Kent, WA 98031 (071)

SAVE YOUR EYES AND YOUR MONEY! Convert your TRS-80 to a Green Screen with this amazing bargain rate conversion kit. Improves image contrast, reduces glare, reduces eye fatigue. Kit includes: 1 sheet of heavy plastic film, double faced mounting tape and instructions. $4.00 Send check or MO, no COD's. P Hayden 2103 Seventh St., East Meadow, NY 11554 Please add 50¢ for postage. (031)

FOR TRS-80 RS-232 BOARD. FIX INTER- mittent problems due to intermittent PC board connector with a connector brace. Easy to install. Only $4.95. Gunn RS-232 board with two 8-bit in & two 8-bit out ports. $35. Bare board and schematic. Schematic $1.00 (TX residents add 5% tax). Gunn Industries, 704 Franklin Blvd, Austin, TX 78751 (071)

WORD PROCESSOR, PERSONAL FORM letter printer. Adds addresses, inserts 1st name from address into letter body. Global word search, replace. Caps-lock, overstrike, move words between lines. Delete, replace, insert, move, center, right-justify lines. Basic-like line edit, with A,C,D,H,I,K,L,S,X commands. T/D save. System? (16K, WP only) $10 on tape, $12 on disk. D D Hinrichs, 2116 S E 377th, Washougal, WA 98671 (051)

DISC REPAIR. MPI, PERTEC, SHUGART, Tandon, disc drive repair. Flat rate of $35.00 plus parts, add $2.00 shipping. All Systems Go. PO Box 915, 105 W Plant St., Winter Garden, FL 32787 (305) 877-2830 (031)

LEVEL II 16K TAPE SYSTEM OWNERS Model I Shoestring Opportunity, no programming, first come first served, geographical patent rights license by Zip, graceful commercial application. Send one page resume & 2 refs, mature private parties only, must have nothing against cocktail lounges; all others send $1000.00 for partial disclosure: MAX Enterprises, PO Box 19386, San Diego, CA 92119 (031)

GL, A/R, A/P, PAYROLL FOR TRSDOS 1.2 Not Osborn's. Fast ISAM, double entry accounting, full screen and integrated MAI 96 Dothan St., Arlington, MA 02174 (091)

ROMAN CONQUEST FOR 16K LEVEL II Caesar orders you to seize cities, sell slaves, draft centurions and colonize. Can you cope with fierce barbarians, plagues, rebellions and the will of Caesar? $9.95 (cassette) Anthony Targonski, 106 Fifth St., Cambridge, MA 02141

TRS-80 MOD III & MOD II & DISK DRIVES best prices fast UPS delivery Call now! Or write Electronic Service Center PO Drawer 11039, Costa Mesa, CA 92627 (714) 957-2900 (031)

UNIQUE TRS-80 SOFTWARE & BOOKS Disassembled Handbook for TRS-80 all postpaid. Vol 1 5th printing "save one year's study" $10. Vol 2 3rd printing "all you wanted to know" $15. Vol 3 New 1981 "exploring new horizons" $17. Morse code transmit program: cassette/disk $15. Richcraft Engineering Ltd. Box 1065, 1 Wahmeda Industrial Park, Chautauqua Lake, NY 14722 Phone (716) 753-2654 for COD orders (111)

DRAW POKER - SOUND AND GRAPHICS program simulates the quarter game you may have seen in various places of business. But you do not need quarters to play this game. Excellent graphics & sound effects. Required: TRS-80 16K Level II. Only $14.00 from Michael D Simon, 25 Flagship Road, Baltimore, MD 21222 (031)

GREATEST BARGAIN! (1) LEVEL II BASIC, 16K software package: word processor, inventory, data base manager, check balancing, statistics, deposit calculator, stock management, label printer, sales analysis, sort and linked list utility. All 11 programs for $59. (2) Integrated accounting package: A/R, A/P, GL and Payroll. $129 each. $20 for manual alone. For TRSDOS I, TRSDOS II, CBASIC and MBASIC on CP/M. For FREE catalog, send SASE (28¢) to MAI, 96 Dothan St., Arlington, MA 02174 (051)

RELOCATE T-BUG TO HIGH RAM. RUN Basic with T-Bug in RAM. Quick return after RESET. Probe the reserved RAM. Level I. Listing $4.00 David S Robertson 2810 SW Caulder, Des Moines, IA 50321 (031)

DATA BASE MANAGER - IDM - "There are many data base managers available these days, some may do the job for you, others may not. This one is doing a very good job for us." - 80-U.S. Jul '80. Now, new documentation and fastest machine sort. Mod II $199. Mod I 48K $149. Micro Architect Inc., 96 Dothan St., Arlington, MA 02174 (111)

INCREASE WIN PROBABILITY WITH harness racing for TRS-80. We believe this is the most advanced harness racing handicapping program available anywhere. Years of research have gone into the development of the algorithms used. Correctly used, a greater return on your track investment will be yours. Requires minimum 16K Level II. Only $24.95 (add $1. shipping; CA residents add 6½ sales tax). Send check or MO to WelMur of California, Inc PO Box M-11B, Fremont, CA 94537 **Coming soon - DOG Racing** (031)

MOD II INTEGRATED ACCOUNTING package. Full functions GL. A/R prints invoices and statements, supports forward balance and open items. A/P helps to select which vouchers to be paid with total cash management. Payroll features user modifiable tax tables. Print W-2, 941, checks, etc. Fast ISAM. As opposed to most other systems, we use 80 col screen instead of 64 and require 132 col printer (not 80), and TRSDOS 1.2, (not CP/M). A similar version using a different tradename is sold for 1000 each, ours is only $129. Includes disk, postage, 100 page manual, an installation guide, and 1 year updates. Micro Architect Inc, 96 Dothan St., Arlington, MA 02174 (031)

COMPILERS

ACCEL2: Compiler for TRS-80 Oisk BASIC Compiles selected subset to Z80 machine code in all four variable types, compact 1K run-time component controls interpreter to streamline all other statements and functions. Technique minimises code expansion without impairing huge speedups for true double optimisation. Six diagnostic messages, Local/Global options increase compatibility with subject programs. Output save to Oisk, tapes. Professionals note: No royalties on the derived code. It's like having a 100 mhz clock **$88.95**

ACCEL: Compiler for TRS-80 Level II BASIC Same huge speedups as ACCEL2 but in INTEGER variable type only Run-time component just 256 bytes, ideal for graphics, games in 16K. Oeveloped in Britain by Southern Software **44.95**

TSAVE: Writes compiler output to SYSTEM tape . . . **$9.95**

SOFTWARE CPU™

Super STEP: Animated Z80 Programming Models, Oisassembler, Single-step/TRACE modes with inteliigent RAM Window, 5 user-selectable Windows, single and cumulative instruction times in microseconds, Reference Space, much more. Big booklet, a Z80 Software CPU. 16K Level II TRS-80, TBUG required. No. BL-O **$19.95**

Super TLEGS: Relocates TBUG, Super STEP . . . **$9.95**

EMU 02: Animated 6502 Programming Models, Disassembles to 6502 mnemonics. Single-step/TRACE modes, 6502 counterparts to #B, #J, #R, #F and #G commands, fast Cross-interpreter, keyboard scan port with p-instructions DB,EB control, paging in virtual address space, more Big booklet & SYNERTEK card, a 6502 Software CPU
16K Level II TRS-80, TBUG required No. BL-1 . . . . **$24.95**

COLOR COMPUTER

COCOBUG: 6809 Oebugging monitor for TRS-80 Color Computer. Examine, modify memory/CPU registers, place breakpoints, execute single instructions or entire machine language programs in real time Includes 6809 Reference card, runs in 4K **$19.95**

MASTERCARD/VISA
Incl. .75 postage, CA add 6%

ALLEN GELDER SOFTWARE
Box 11721 Main Post Office
San Francisco, CA 94101

TRS-80, TBUG tm Radio Shack/Tandy Corp
Software CPU tm Allen Gelder Software

Circle 78

Let Your TRS-80® Teach You ASSEMBLY LANGUAGE

Tired of buying book after book on assembly language programming and still not knowing your **POP** from your **PUSH?**

REMSOFT proudly announces a more efficient way, using your own TRS-80® , to learn the fundamentals of assembly language programming --at YOUR pace and at YOUR convenience.

Our unique package, "**INTRODUCTION TO TRS-80® ASSEMBLY PROGRAMMING**", will provide you with the following:

- Ten 45-minute lessons on audio cassettes.
- A driver program to make your TRS-80® video monitor serve as a blackboard for the instructor.
- A display program for each lesson to provide illustration and reinforcement for what you are hearing.
- A textbook on TRS-80® Assembly Language Programming.
- Step-by-step dissection of complete and useful routines to test memory and to gain direct control over the keyboard, video monitor, and printer.
- How to access and use powerful routines in your Level II ROM.

This course was developed and recorded by Joseph E. Willis and is based on the successful series of courses he has taught at Meta Technologies Corporation, the Radio Shack Computer Center, and other locations in Northern Ohio. The minimum system required is a Level II, 16K RAM.

REMASSEM-1 only $69.95

REMSOFT, Inc.
571 E. 185 st.
Euclid, Ohio 44119
(216)531-1338

Include $1 50 for shipping and handling.
Ohio residents add 5½% sales tax.
TRS-80® is a trademark of the Tandy Corp.

Circle 79

AT-80 ANNOUNCES A NEWDOS SPECIAL

DDIR80 — Creates program lines of NEWDOS DIR's, adding them to itself. Options include — Search, Re-search, Run, Hardcopy, Display DIR's, Others. Stores up to 175 DIR's. 32K/one disk. $23, w/demo.

CAT — Tic-Tac-Toe with randomly numbered squares. FAST graphics. Human vs human option. $12.

FTDEMO80 — Displays the programs, and the keyboard commands, from the NEWDOS/80 Appendix A examples, *WHILE* executing the programs and commands and displaying results. Cycle through the five file types with only the enter key. $12.

Disk only. Deduct $3 each for 2nd and 3rd program ordered.

NEWDOS $45 NEWDOS+ $95
NEWDOS/80 $145
Add 4% for MasterCard/Visa

AT-80
3827 Dismount
Dallas, TX 75211
(214) 339-0498

Circle 81

MODEL I or III SOFTWARE

intraSORT *machine code* becomes part of **YOUR BASIC PROGRAM.**

- CLOAD CSAVE LOAD SAVE **TAPE or DISK**
- **NO MACHINE LANGUAGE KNOWLEDGE REQUIRED**
- NO **SYSTEM or MEMORY SIZE** hassles
- **SORT** *any* type and combination of fields with each field ascending or descending
- **SORT** multi-dimension arrays
- **VERY FAST** and compact
- Complete USERS GUIDE
- Includes full feature mailing list/label program demonstrating all **SORT** features
- ••• COMPLETE PACKAGE ONLY $19.95 (requires Level II or Model III Basic, 16K)

BASIC ARCADE KIT *(Level I only)*

Hours of action fun for adults and children are provided by a single ready to run program written entirely in Level I BASIC. Designed for the novice programmer, twelve easy to use, fully documented subroutines and three demo games; TAG, PHASERS, and SKETCH make up the program. Subroutines provide: three keyboard commands which alter graphics motion without halting it; keyed input without disruption of display; two players with separate skill levels & scoring; obstacle deflection; random motion; and cassette storage of graphic displays. Write your own games or just ENJOY!

includes 20 page guide ONLY $9.95

Original Software by:

Circle 80

technique COMPUTER SOFTWARE
2720 S. Oak St • Port Angeles, WA 98362

Circle 82

AT LAST!

Mass production prices for high quality software. Buy direct and save 50%. Also available for CPM and HDOS.

DATA BASE MANAGER Mod I & III $69, $149 (48K). Mod-II $199
Maintain a data base and produce reports, all without user programming. Define file parameters and report formats on-line. Key random access, fast multi-key sort, field arithmetics, audit log, label. No time-consuming overlays. 500 happy users in one year. Mod-II and 48K versions have over 50 enhancements, including 40 fields maximum. "IDM-M2 is great!" - 80-US.

A/R Mod-I $69 Mod-II $149 Mod-III $69
Handles invoices, statements, aging, sales analysis, credit checking, forms input, and order entry. Unlike other accounts receivable programs, ours can be used by doctors, store managers, etc.

WORD PROCESSOR $49
Centers, justifies, indents, and numbers pages. Mod-I version features upper/lower case without hardware modification! File merge option available.

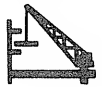
MAILING LIST Mod I & III $59, $79 (48K). Mod-II $99
The best! Compare and be selective. Includes forms input, 5-digit selection code, zip code extension, sort on any field, and multiple labels. Who else offers a report writer and merges with word processor?

INVENTORY Mod I & III $89, $109 (48K) Mod-II $149
Fast key random access. Reports include order info, performance summary, EOQ and user-specified reports. Many people have converted to our system! "Next to impossible to damage the file."

GL. A/R, A/P, PAYROLL Mod-II $129 each
Integrated accounting package. 100+ page manual. As opposed to Osborne's slow binary search and 64 column screen, we use fast ISAM and 80 columns. Dual disk and TRSDOS required.

L216 $59
A cassette package of 10 business programs for Level II 16 K systems. Includes word processor and data base manager. Poker game $19.

Most programs are on-line, interactive, random-access, bug-free, documented, and delivered on disks. Mod-I programs require 32K TRSDOS. We're #1 in business software—don't let our low price fool you! Ask for our free 20-page catalog if you're still not convinced. Compiled versions are available.

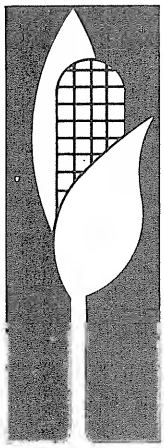


MICRO ARCHITECT, INC.
96 Dothan St., Arlington, MA 02174

BACK ISSUE AVAILABILITY

The following back issues of 80-U S are still available:
Jul/Aug 1979
Nov/Dec 1979
Jan/Feb 1980
May/Jun 1980
Nov/Dec 1980
Jan/Feb 1981

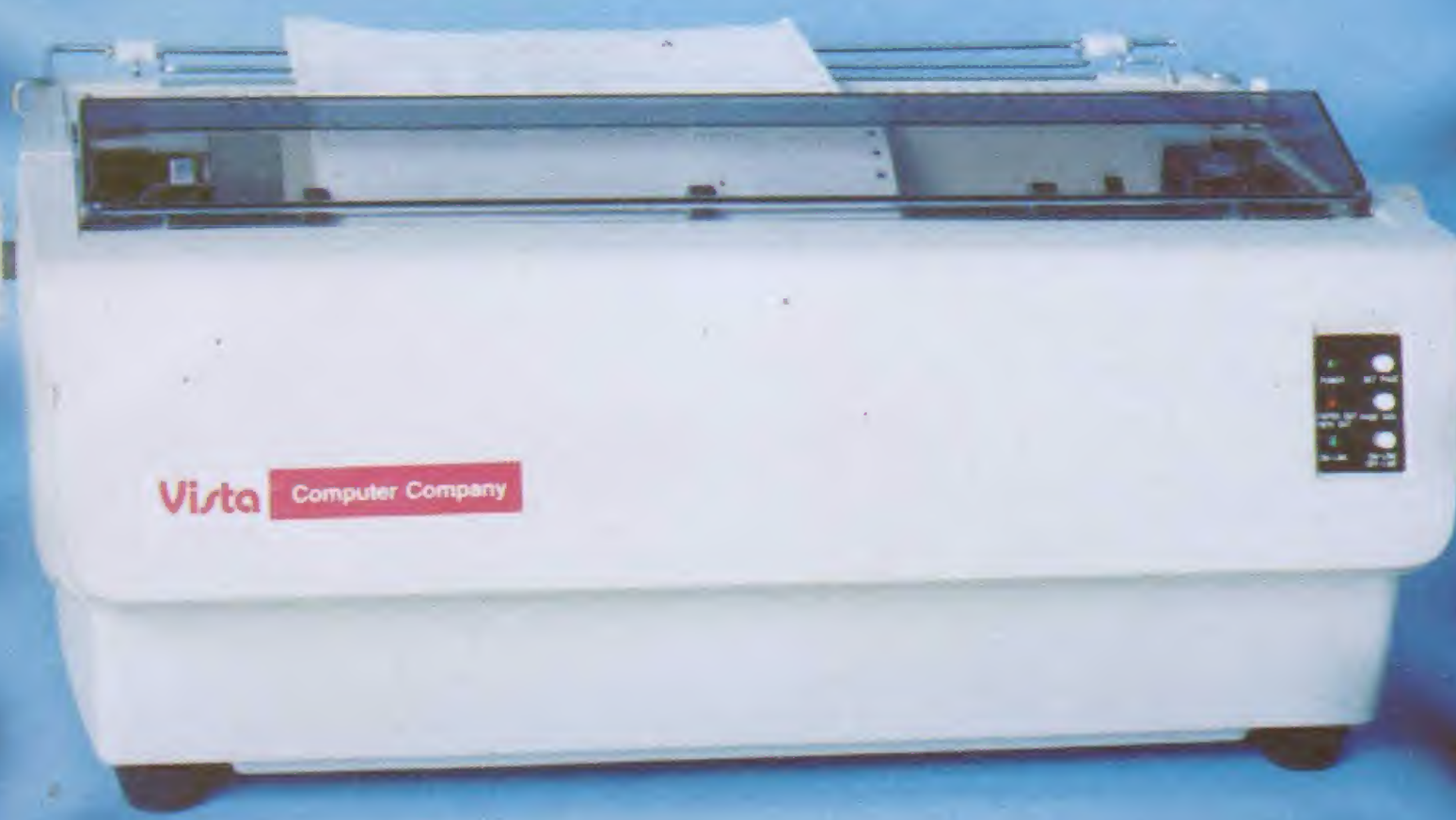
Price per copy is $3.00 Please allow 3 to 4 weeks for delivery.



Circle 83

ADVERTISER & READER SERVICE INDEX

When you pick a Daisy...
Pick Vista's V300 Printer.



$1895.

There are Daisies!. . . And, There are Daisies!. . . But Vista has a Peach!

The Vista V300 is exactly that, a "peach" of a daisy wheel printer both from the standpoint of price and performance.

Think of it, a printer at nearly half the price (when compared to models even remotely competitive in quality) combined with the ultimate in reliability, print quality, and flexibility.

Typical Comments: "Superb print quality!", "Highly reliable.", "Definitely letter quality. . . I can't believe the price tag.", "Best use I've seen yet of LSI Technology."

But judge for yourself — look at the V300 features and keep in mind this is a letter quality printer at dot matrix prices.

- **Tractor option available**
- **Print Speed — 25 CPS (Optional 45 CPS for $2,195)**
- **Print Wheel** — Industry standard 96-character Daisy Wheel (including the extended-life dual plastic wheels)
- **Service** — Prompt maintenance/service agreements available nationwide
- **Interface** — Industry standard parallel (RS232-C optional)
- **Printable Columns** — 136
- **Warranty — 90 days parts and labor, one year parts only**
- Proportional, bi-directional printing • Programmable VFU
- Extensive self-test functions • Hardware and software compatible

Vista does it again! Quality, Price and Performance with a peach of a daisy wheel printer.

AND, Vista Has a Complete V100 Word Processing System for Only $4995!

The Vista V100 is a complete word processing system that includes:

- Exidy Sorcerer Computer, 48K
- V200E20 Disc Drive System, Double Density
- Sanyo Data Display Monitor
- Vista V300 Printer Full Character Daisy Wheel
- Wordstar, CPM 1.4 (Includes E Basic)
- Can also be used for Data Processing

Vista

IMMEDIATE DELIVERY

For Further Information
Call Toll Free (800) 854-8017

The Vista Computer Company 1401 Borchard Street • Santa Ana, California 92705 • 714/953-0523

Pump Up Your TRS-80 with the ES/F Mass Storage System

▲ Actual Size

Actual Thickness ▼

THESE FACTS SPEAK FOR THEMSELVES!

	CASSETTE	ES/F	MINI-DISK
SPEED (Seconds to load "Blackjack")	56	6 (5' wafer)	6½
CAPACITY (thousands of bytes)	38 (C-20)	64 (75' wafer)	59 (TRSDOS)
RELIABILITY (Designed for digital data?)	NO	YES	YES
SYSTEM COST (First unit plus interface)	$60	$250	$800
MEDIA COST (in quantities of ten)	$3.10 cassette	$3.00 wafer	$3.20 disk

Let's face it. Cassette players were not designed to store digital data and programs. That's why we designed a digital storage system using a continuous tape loop: the Exatron Stringy/Floppy (ES/F) and the Wafer. There's no expensive interface to buy—the ES/F comes ready to pump up your TRS-80.*

Once your TRS-80* is pumped up by our ES/F . . . you won't want to deflate it. We're so sure, that we offer an unconditional 30-day money-back guarantee and a one-year limited warranty. Over 2,000 TRS-80* owners have met the wafer . . . why don't you?

TRS-80*

EXATRON'S STRINGY/FLOPPY...

SPEED, CAPACITY AND RELIABILITY FOR ONLY $249.50

CALL
OUR HOTLINE
(800)-538-8559

IN CALIFORNIA,
CALL (408)-737-7111

exatron, inc.
181 Commercial Street
Sunnyvale, Calif. 94086

*TRS-80 is a registered trademark of Tandy Corp.